Confidence Scoring Using Whitebox Meta-models with Linear Classifier Probes

Tongfei Chen[†]   **Jiří Navrátil**[‡]   **Vijay Iyengar**[‡]   **Karthikeyan Shanmugam**[‡]

[†] Johns Hopkins University

[‡] IBM Thomas J. Watson Research Center

Abstract

We propose a novel confidence scoring mechanism for deep neural networks based on a two-model paradigm involving a *base model* and a *meta-model*. The confidence score is learned by the meta-model observing the base model succeeding/failing at its task. As features to the meta-model, we investigate linear classifier probes inserted between the various layers of the base model. Our experiments demonstrate that this approach outperforms multiple baselines in a filtering task, i.e., task of rejecting samples with low confidence. Experimental results are presented using CIFAR-10 and CIFAR-100 dataset with and without added noise. We discuss the importance of confidence scoring to bridge the gap between experimental and real-world applications.

1 Introduction

With the advancement of deep learning techniques, models based on neural networks are entrusted with various applications that involve complex decision making, such as medical diagnosis (Caruana et al., 2015), self-driving cars (Bojarski et al., 2016), or safe exploration of an agent's environment in a reinforcement learning setting (Kahn et al., 2017). While the accuracy of these techniques has improved significantly in recent years, they are lacking a very important feature: an ability to reliably detect whether the model has produced an incorrect prediction. This is especially crucial in real-world decision making systems: if the model is able to sense that its prediction is likely incorrect, control of the system should be passed to fall-back systems

or to a human expert. For example, control should be passed to a human medical doctor when the confidence of a diagnosis with respect to a particular symptom is low (Jiang et al., 2011). Similarly, when a self-driving car's obstruent detector is not sufficiently certain, the car should rely on fall-back sensors, or choose a conservative action of slowing down the vehicle (Kendall and Gal, 2017). Lack of, or poor confidence estimates may result in loss of human life (National Highway Traffic Safety Administration, 2017).

We address this problem by pursuing the following paradigm: a learnable confidence scorer acting as an "observer" (**meta-model**) on top of an existing neural classifier (**base model**). The observer collects various features from the base model and is trained to predict success or failure of the base model with respect to its original task (e.g., image recognition).

Formally, a meta-model G that, given a base model $y = F(x)$, should produce a confidence score $z = G(x, \Theta_F)$ (where Θ_F denotes the parameters of the base model F). The confidence score z need not be a probability: it can be any scalar value that relates to uncertainty and can be used to filter out the most uncertain samples based on a threshold value.

To generate confidence scores, we propose a meta-model utilizing *linear classifier probes* (Alain and Bengio, 2016) inserted into the intermediate layers of the base model (hence referred to as "*whitebox*" due to its transparency of the internal states). We use a well-studied task, image classification, as the focus of this paper, and show that the confidence scores generated by the whitebox meta-models are superior to standard baselines when noisy data are considered in the training. By removing samples deemed most uncertain by our method, the precision of the predictions by the base model on the remaining examples improves significantly. Additionally, we show in the experiments that our method extends to handling out-of-domain samples: when the base model encounters out-of-domain data, the whitebox meta-model is shown to be capable of rejecting these with better accuracy than baselines.

2   Related work

Previous work on Monte Carlo dropout (Gal et al., 2017; Gal and Ghahramani, 2016) to estimate model uncertainty can be applied to our filtering task at hand. In an autonomous driving application this approach showed that model uncertainty correlates with positional error (Kendall and Cipolla, 2016). In an application to image segmentation, uncertainty analysis was done at the pixel level and overall classification accuracy improved when pixels with higher uncertainty were dropped (Kampffmeyer et al., 2016). Monte Carlo dropout was also used to estimate uncertainty in diagnosing diabetic retinopathy from fundus images (Leibig et al., 2017). Diagnostic performance improvement was reported when uncertainty was used to filter out some instances from model based classification.

Uncertainty estimations from methods like Monte Carlo dropout can be viewed as providing additional features about a model's prediction for an instance, which can be subsumed by our proposed meta-model approach.

In a broader context, the ability to rank samples is a fundamental notion in the *receiver operating characteristics* (ROC) analysis. ROC is primarily concerned with the task of detection (*filtering*) which is in contrast to estimating a prognostic measure of uncertainty (implying *calibration*). Plethora of ROC-related work spanning a variety of disciplines, including biomedical, signal, speech, language, and image processing, has been explored in the context of filtering and decision making (Zou et al., 2011; ICML Workshop). Moreover, ROC, either as a whole or through a part of its operating range, has been used in optimization in various applications (Wang et al., 2016; Navrátil and Ramaswamy, 2002). Since we are focusing on the filtering aspect of confidence scoring rather than their calibration, we adopt the ROC analysis as our primary metric in this work (Ferri et al., 2009).

Modern neural networks are known to be miscalibrated (Guo et al., 2017): the predicted probability is highly biased with respect to the true correctness likelihood. Calibration has been proposed as a postprocessing step to mitigate this problem for any model (Zadrozny and Elkan, 2001, 2002; Guo et al., 2017). Calibration methods like isotonic regression (Zadrozny and Elkan, 2002) perform transformations that are monotonic with respect to scores for sets of instances and so will not alter the ranking of confident vs. uncertain samples. The more recent temperature scaling calibration method (Guo et al., 2017) can alter the ranking of instances and will be considered and compared in our analysis.

The recent work on selective classification for deep neural networks (Geifman and El-Yaniv, 2017) shares the same broad goals to filter out instances where the base model prediction is in doubt. Their method uses only the outputs of the base model (*softmax response*) to determine a threshold that would optimize coverage (recall) while guaranteeing the desired risk (precision) at some specified confidence level. From an application perspective, our work extends this by showing that in noisy settings whitebox models for this task outperform methods using only the base model output scores. We also consider an additional task using out-of-domain instances to evaluate filtering methods when encountering domain shifts.

3   Method

For any classification model $\hat{\mathbf{y}} = F(\mathbf{x})$ where $\hat{\mathbf{y}}$ is the probability vector of the predicted classes, we define a confidence scoring model (G, the *meta-model*) that operates on F (*base model*) and produces a confidence score z for each prediction $\hat{\mathbf{y}}$.

We explore two kinds of meta-models, namely the *blackbox* and the *whitebox* type.

Blackbox   In the blackbox version it is assumed that the internal mechanism of the model F is not accessible to the meta-model, i.e., the only observable variable for the meta-model is its output $\hat{\mathbf{y}}$:

$$z = G_{blackbox}(\hat{\mathbf{y}}). \qquad (1)$$

For example, in a k-class classification problem, the meta-model is only allowed to take the final k-dimensional probability vector into account. A typical representative of a blackbox baseline commonly employed in real-world scenarios is the *softmax response* (Geifman and El-Yaniv, 2017): just taking the probability output of the predicted class label:

$$z = P(y^*|\mathbf{x}, \boldsymbol{\Theta}_F) = \max_i \hat{\mathbf{y}}_{(i)}, \qquad (2)$$

where $\hat{\mathbf{y}}_{(i)}$ is the i-th dimension of the vector $\hat{\mathbf{y}}$, $y^* = \arg\max_i \hat{\mathbf{y}}_{(i)}$ (i.e. the label with the highest predicted probability), and $\boldsymbol{\Theta}_F$ denotes the parameters of the base model F.

Whitebox   A whitebox meta-model assumes full access to the internals of the base model. A neural model, consisting of multiple stacked layers, can be regarded as a composition of functions:

$$F(\mathbf{x}) = f_n(f_{n-1}(\cdots(f_2(f_1(\mathbf{x})))\cdots)). \qquad (3)$$

We denote the intermediate results as $\mathbf{x}_1 = f_1(\mathbf{x})$; $\mathbf{x}_2 = f_2(\mathbf{x}_1)$; $\mathbf{x}_3 = f_3(\mathbf{x}_2)$; etc. A *whitebox* meta-model is capable of accessing these intermediate results:

$$z = G_{whitebox}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n), \qquad (4)$$
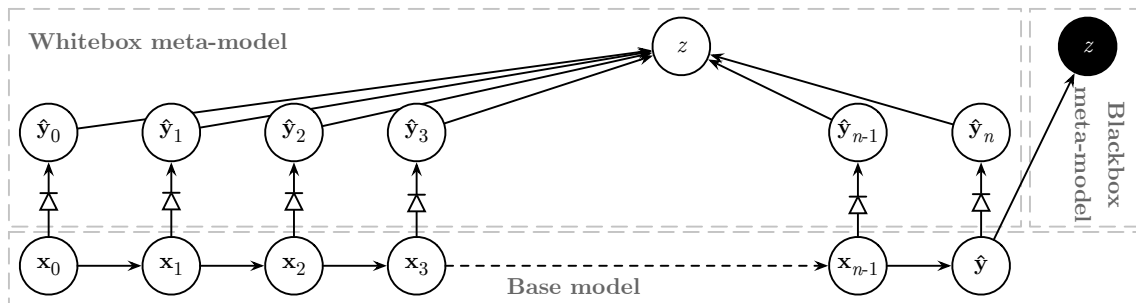
Figure 1: A schematic overview of whitebox vs. blackbox meta-models.

where $\mathbf{x}_n = \hat{\mathbf{y}}$ is the output of the last layer. It should be noted that in general the meta-model may employ additional functions to combine the base model's intermediate results in various ways, and we explore one such option by using linear classifier probes described below.

3.1 Whitebox meta-model with linear classifier probes

We propose a whitebox model using linear classifier probes (later just "*probes*"). The concept of probes was originally proposed by (Alain and Bengio, 2016) as an aid for enhancing the interpretability of neural networks. However, we are applying this concept for the purpose of extracting features from the base model. Our intuition draws from the fact that probes for different layers tend to learn different levels of abstractions of the input data: lower layers (those closer to the input) learn more elementary patterns whereas higher layers (those closer to the output) capture conceptual abstractions of the data and tend to be more informative with respect to the class label of a given instance.

For each intermediate result $\mathbf{x}_i$ ($0 < i \leq n$ with $\mathbf{x}_n = \hat{\mathbf{y}}$ being the final output of a multi-layer neural network), we train a probe $F_i(\mathbf{x}_i)$ to predict the correct class y using only the specific intermediate result:

$$\hat{\mathbf{y}}_i = F_i(\mathbf{x}_i) = \text{softmax}(\mathbf{W}_i \mathbf{x}_i + \mathbf{b}_i) . \qquad (5)$$

Given a set of trained probes, $\{F_i\}_{0<i\leq n}$, we build the meta-model using the probe outputs (either probabilities or logits) as training input. The meta-model is then trained with the objective of predicting whether the base model's classification is correct or not. Finally, the prediction probability of the base model being correct is the confidence score z:

$$z = G(\hat{\mathbf{y}}_1, \cdots, \hat{\mathbf{y}}_n). \qquad (6)$$

This architecture is illustrated in Figure 1. The diode symbol " —▶— " represents the one-way nature of the

information flow emphasizing that the probes are not trained jointly with the base model. Instead, they are trained with the underlying base model's parameters fixed.

3.2 Meta-model structure

We explore three different forms of the meta-model function G from Eq. (6). The meta-model is trained as a binary classifier where G predicts whether the base model prediction is correct or not. The probability of the positive class $P(\text{"base model is correct"}|\hat{\mathbf{y}}_1, \cdots, \hat{\mathbf{y}}_n, \boldsymbol{\Theta}_G)$ is used as the confidence score z.

Logistic regression (LR) This meta-model has a simple form

$$z = \frac{e^s}{1 + e^s} \text{ with } s = \boldsymbol{\theta} \cdot [\hat{\mathbf{y}}_1 \ \hat{\mathbf{y}}_2 \ \cdots \ \hat{\mathbf{y}}_n]. \qquad (7)$$

where the probe vectors $\hat{\mathbf{y}}_i$ are concatenated. The logit value $z \in (0, 1)$ in Eq. (7) is used directly as the confidence score. The model is L_2-regularized.

Gradient boosting machine (GBM) The concatenated probe vectors are fed into a gradient boosting machine (Friedman, 2001). The GBM hyperparameters include the learning rate, number of boosting stages, maximum depth of trees and the fraction of samples used for fitting individual base learners.

Besides the aforementioned structures, we also investigated fully connected 2-layer neural networks, however, omitted them in this paper as their performance was essentially identical with the GBMs.

4 Tasks, datasets and metrics

We use the CIFAR-10 and CIFAR-100 image classification dataset[1] in our experiments. For each set of data we conduct two flavors of experiments: In-domain

---

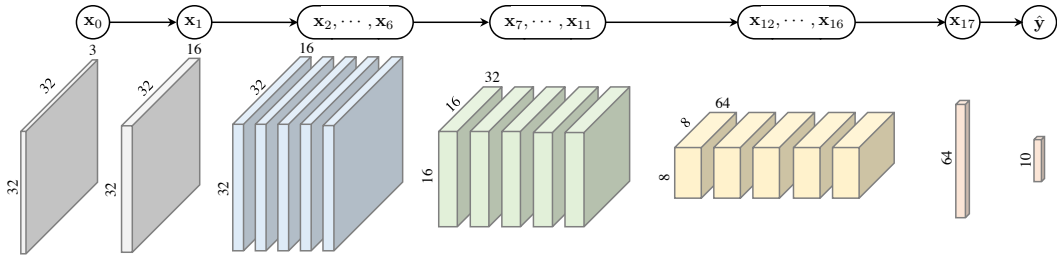[1] https://www.cs.toronto.edu/~kriz/cifar.html.

Figure 2: Neural structure of the base model.

confidence scoring task and an in-domain plus out-of-domain pool task (referred to as "out-of-domain" from now on).

In-domain task Given a base model and a held-out set, the base model makes predictions about samples in the held-out set. Can the trained meta-model prune out predictions considered uncertain? Furthermore, after removing a varying percentile of the most uncertain predictions, how does the residual precision of the pruned held-out set change? The expected behavior is that the proposed meta-model should increase the overall residual accuracy after uncertain samples are removed.

Out-of-domain task Given a base model (here trained on CIFAR-10), what would the model do if presented with images not belonging to one of the 10 classes? The predictions made by the base model will surely be wrong: However, can the meta-model distinguish these predictions as incorrect? Our proposed meta-model should in theory produce a low confidence score to these out-of-domain predictions. Note that the out-of-domain task comprises both in-domain and out-of-domain samples to be processed as a single pool.

We use the ROC (receiver operating characteristic) curve and the precision/recall curve to study the diagnostic ability of our meta-models. In the ROC curve, the x-axis is the false positive rate (i.e. rate of incorrectly detected success events) and the y-axis is the true positive rate (i.e. recall): a operating point (x_0, y_0) on the ROC plot corresponds to threshold inducing a trade-off between a proportion of x_0 wrongly classified samples not detected by the meta-model and y_0 proportion of correctly classified samples that the meta-model agrees with.

Additionally, we compute the area under curve (AUC) for the ROC curve as a summary value.

4.1 Datasets

The original CIFAR-10 dataset contains 50,000 training images and 10,000 test images. We divide the original

training set into 3 subsets, namely TRAIN-BASE, TRAIN-META and DEV.

Table 1: Dividing the CIFAR-10 dataset.

Original partition	New partition	Size
50,000 train	TRAIN-BASE	30,000
	TRAIN-META	10,000
	DEV	10,000
10,000 test	TEST	10,000

We adopt the following training strategy, so as to completely separate the data used by the base model and the meta-model:

- Train the base model using the TRAIN-BASE subset: Because the size of the training set is smaller (30,000 samples instead of 50,000) than the standard setup (reported as 92.5% accuracy using the base model), the accuracy on DEV and TEST is slightly lower: we get 90.4% accuracy on TEST.

- Train the whitebox meta-model (including the probes) on TRAIN-META.

- The DEV set is used for tuning (various hyperparameters) and for validation.

- The TEST set is used for final held-out performance reporting.

The out-of-domain task is evaluated by combining the test sets of CIFAR-10 and CIFAR-100 datasets. The CIFAR-100 dataset class labels are completely disjoint with those of CIFAR-10. The out-of-domain set will be referred to as OOD.

4.2 Base model

We reuse the high performing ResNet model for image classification implemented in the official TensorFlow

(Abadi et al., 2016) example model code[2]. This model consists of a sequential stack of residual units of convolution networks (He et al., 2016a,b; Zagoruyko and Komodakis, 2016) as shown in Figure 2. Each layer's tensor size is specified in the figure.

In subsequent experiments, we train probes for all intermediate layers[3] from $\mathbf{x}_1$ to $\hat{\mathbf{y}}$.

5 Experimental results

To assess the various models we organize the experiments in several parts by varying the quality of the data used to create the models. Furthermore, their performance in each part is evaluated on both the *in-domain* and the *out-of-domain* tasks. The varying quality aspect comprises the following conditions:

Clean base / Clean meta All sets involved in training, i.e., TRAIN-BASE, TRAIN-META, and DEV are used in their original form from the CIFAR-10 dataset;

Noisy base / Noisy meta In this case the sets TRAIN-META and DEV are modified by adding artificial noise to the labels of the images, hence degrading the base model performance. Specifically, for a random subset of 30% of the samples, the correct label is replaced by another label (randomly chosen over the corresponding complement of the label set). This results in an artificially degraded base model with a test set accuracy of 77.4% (as compared to 90.4% of the same model trained on clean data). This condition, in combination with the degraded base model, represents a scenario of obtaining training data from a noisy environment, e.g., via crowd-sourcing in which labels are not always correct.

In both conditions, the TEST set (both *in-domain* and *out-of-domain*) is applied clean, without artificial corruption. The above conditions in combination with the two tasks offer a representative set of classification scenarios encountered in practice.

We compare the following confidence scoring methods:

- (**Softmax**) Softmax response function (Eq. (2)) in Geifman and El-Yaniv (2017);

- (**Blackbox-LR/GBM**) Using the final output $\hat{\mathbf{y}}$ as the only feature for the meta-models;

- (**Probes-LR/GBM**) Whitebox model using all the probes as features for the meta-models.

---

[2] https://github.com/tensorflow/models/tree/master/research/resnet.
[3] We do not insert probes between the two convolutional layers within the residual unit, instead, we consider a residual unit as an atomic layer.

Table 2: Area under ROC (AUC) of various methods.

Method	Condition (base/meta)	
	Clean/Clean	Noisy/Noisy
In-domain Tasks		
Softmax	**0.91**	0.74
Blackbox (LR)	**0.91**	0.79
Blackbox (GBM)	**0.91**	0.80
Probes (LR)	0.88	0.87
Probes (GBM)	**0.91**	**0.88**
Out-of-domain Tasks		
Softmax	**0.89**	0.72
Blackbox (LR)	**0.89**	0.76
Blackbox (GBM)	**0.89**	0.77
Probes (LR)	0.85	0.83
Probes (GBM)	**0.89**	**0.84**

Fig. 3 shows the main results for the two conditions and two datasets defined above, in terms of ROC and Precision/Recall curves. Table 2 summarizes the AUC (area under ROC) results.

Under the Clean/Clean condition we observe little difference among the methods, with AUC values at 0.91 (in-domain setting for the test set, later on, TEST) and 0.89 (out-of-domain setting, later on, OOD) (with the exception of the Probes-LR model, see discussion below).

On the other hand, under Noisy/Noisy condition, the probe-based (whitebox) models separate themselves well from the baseline as well as their blackbox counterparts. Under the Noisy/Noisy condition, the Probes-GBM model with AUC values of 0.88 (TEST) and 0.84 (OOD) dominates its Blackbox-GBM counterpart at 0.80 (TEST) and 0.77 (OOD).

Overall, under the Noisy/Noisy condition, two trends can be identified: (1) whitebox probe-based models outperform their blackbox counterparts, all of which fare significantly better than the softmax baseline, and (2) the probe-based GBM model dominates, albeit moderately, the simpler LR model in all cases.

We analyzed further the lower performance of the L_2-regularized Probes-LR model in the Clean/Clean condition. We explored variants including a sparse L_1-regularized LR model but could not find a satisfactory answer to this performance drop.

We also compared the performance of the temperature scaled base model scores (Guo et al., 2017) in the two cases, Clean/Clean and Noisy/Noisy: The performances for both in-domain and out-of-domain tasks after scaling when compared to the original base model scores stayed essentially the same in each case,
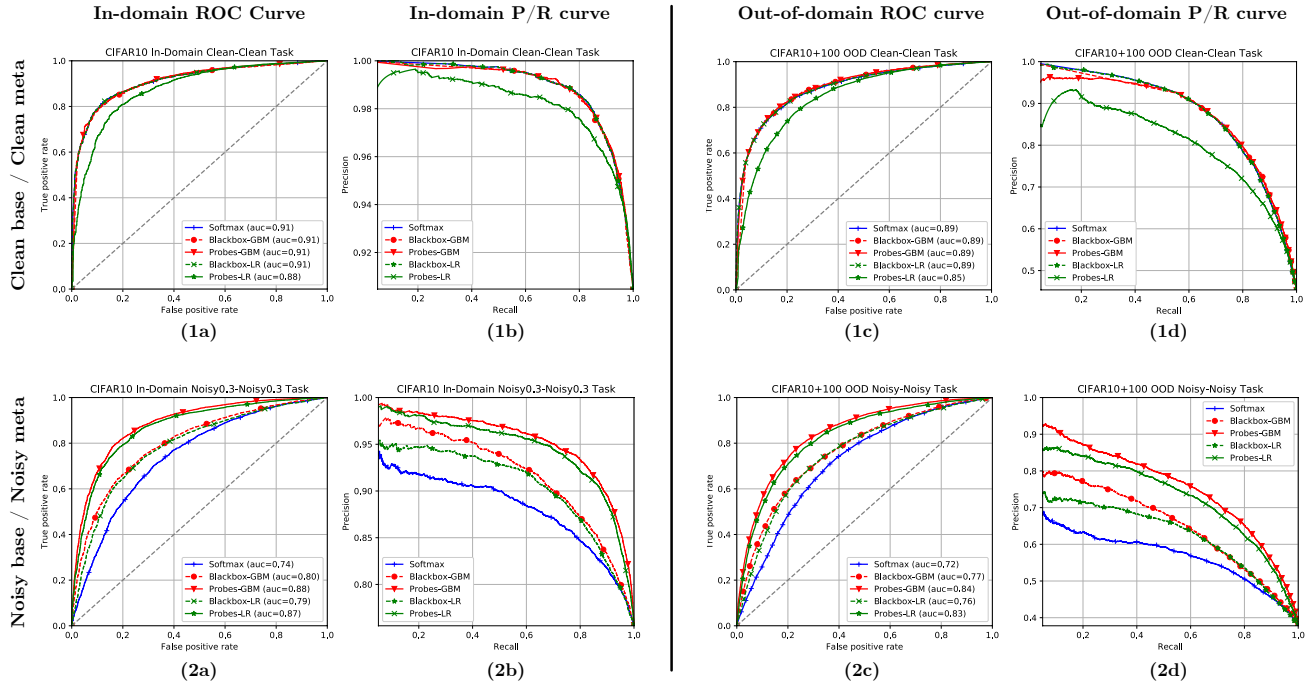
	In-domain ROC Curve	In-domain P/R curve		Out-of-domain ROC curve	Out-of-domain P/R curve

Figure 3: Figures (1a)-(1d) show the performance metrics for the various model in the "Clean/Clean" condition, i.e., when both the base model as well as the meta-model were trained using uncorrupted data. The AUC (area under curve) values were calculated for each model and are shown in the corresponding legend of the ROC plots. Performance curves for the "Noisy/Noisy" condition, i.e., one where both the base and the meta-model are degraded by 30%-noise, are shown in Figures (2a)-(2d).

suggesting that the task of calibration remains an *orthogonal* aspect of confidence scoring (i.e., changing the distribution of the predicted scores but not sample ranking).

6   Discussion

The experimental results presented in the previous section show that whitebox meta-models using probes are significantly better in noisy settings and also in out-of-domain settings when compared to softmax baseline and blackbox models, as is shown by the various ROC or precision/recall curve plots. In this section we will extract some insights by diving deeper into the results.

It is instructive to start with a comparison of accuracies achieved by the probes at various levels. The chart in Figure 4 depicts these accuracies based on the meta-model training data in the two scenarios: Clean base / Clean meta, Noisy base / Noisy meta, respectively. The impact of noise is seen in the top accuracy achieved in one of the two scenarios. The accuracy improves with neural network depth for the most part in both scenarios. We also explored non-linear probes using neural networks with one hidden layer of size 100.

Although the probe accuracies did improve for many of the earlier layers the resulting meta model performance remained comparable and therefore we present results using the simpler linear probes only.

The accuracy plots do not provide insights into how the whitebox models achieve their higher performance and
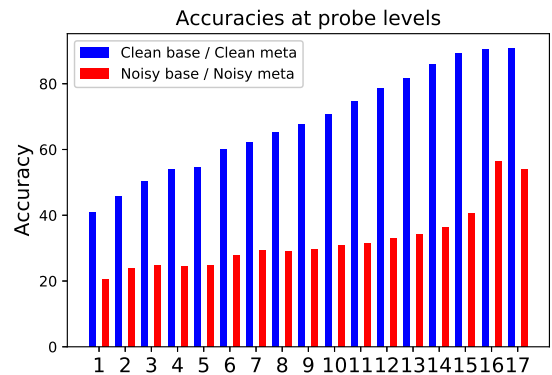

Figure 4: Accuracies for the meta-model training data at varying probe levels.

how this changes going from the clean data scenario to the scenarios with added label noise.

To gain additional insight we performed a feature informativeness analysis based on a method described in (Friedman, 2001). Derived from the GBM meta-model's feature usage statistics using the test set, feature importance scores for the two conditions (Clean/Clean and Noisy/Noisy) are shown in Figure 5. Here, each of the 10 outputs of each of the 17 probes is assigned an intensity level according it its importance score, thus forming a heatmap representation. Recall that the features are sorted according to the top-layer class probabilities, i.e, for each sample, feature 1 (on the vertical axis in Figure 5) corresponds to the top-scoring class, feature 2 to 2-nd highest scoring class, etc., across all the probes (horizontal axis).

Considering the Clean/Clean scenario first (top portion in Figure 5), the top important features include probe outputs in the last layer (Layer 17), focusing on the score of the predicted class (i.e., output with the highest base model score) and the class with the second highest base model score. This aligns with the intuition that having a high score for the predicted class and a large gap relative to the next competing class (i.e., mostly looking at top 2 scores) is indicative of the base model being correct. However, the observation changes in the Noisy/Noisy scenario (bottom portion of Figure 5). Here, two observations can be made: (1) there is a distinct shift in reliance of the GBM on the second-to-last layer (Layer 16), preserving the pattern of looking at the top 2-3 scores within the probe, and (2) a significantly deeper-reaching attention of the meta-model within the probe cascade, including layers 12 through 16. We conjecture that these observations reflect the meta-model's pattern of "hedging" against the adverse effect of the label noise introduced in the Noisy-Noisy task. As the base model's error rate becomes higher (approximately 25%), the meta-model learns to almost completely ignore the Layer 17 (which is directly exposed to the label noise) and to pick up on more robust, deeper-residing features in the ResNet model. This ability to adjust is the powerful advantage of the meta-model approach and seems to lead to its significant performance improvement in the noisy scenario.

There is another advantage of the whitebox meta-models that can be illustrated by considering the relative performance in the in-domain and out-of-domain settings. We argue that the Noisy/Noisy scenario is relevant for many real-life applications in which labels for the training data come from noisy sources. Figure 6 shows the comparative performances in in-domain and out-of-domain settings for the whitebox GBM meta-model and the base model final scores, respectively.
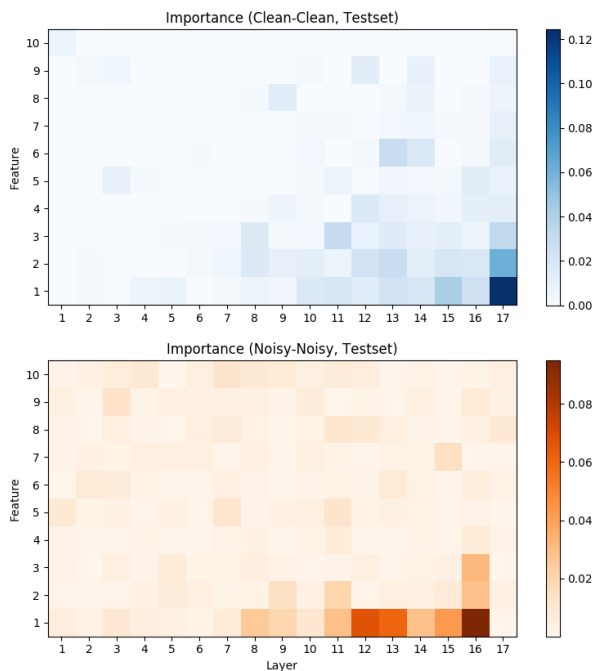

Figure 5: Feature importance scores as obtained on the Testset in Clean-Clean (top) and Noisy-Noisy (bottom) conditions, arranged by their corresponding probe layer and ordered position in each probe.

The x-axes in these plots represents the corresponding threshold values for the respective models for filtering the base model predictions (i.e., samples with confidence scores lower than the threshold value would be filtered). First, consider the whitebox meta-model case in Figure 6 (left). Let's say, in an application setting, we pick a threshold (≈ 0.59) that achieves an in-domain recall of 0.7. At this threshold, the GBM whitebox meta-model achieves an in-domain precision of 0.95. If we encounter a domain shift as represented by the out-of-domain task the precision degrades to ≈ 0.71. Consider the same situation if we were using the base model score as in Figure 6 (right). The threshold value of ≈ 0.51 achieves the same in-domain recall of 0.7. The in-domain precision is 0.87 but the drop in precision for the out-of-domain case is steeper to ≈ 0.54. The lower performance degradation for whitebox meta-models when encountering domain shifts can be viewed as a form of robustness when compared with simply using the base model's scores.

The impact of meta-model based filtering can be further illustrated using examples representing four quadrants of the binary confusion matrix. We chose the CIFAR-10 class "deer" and considered all instances
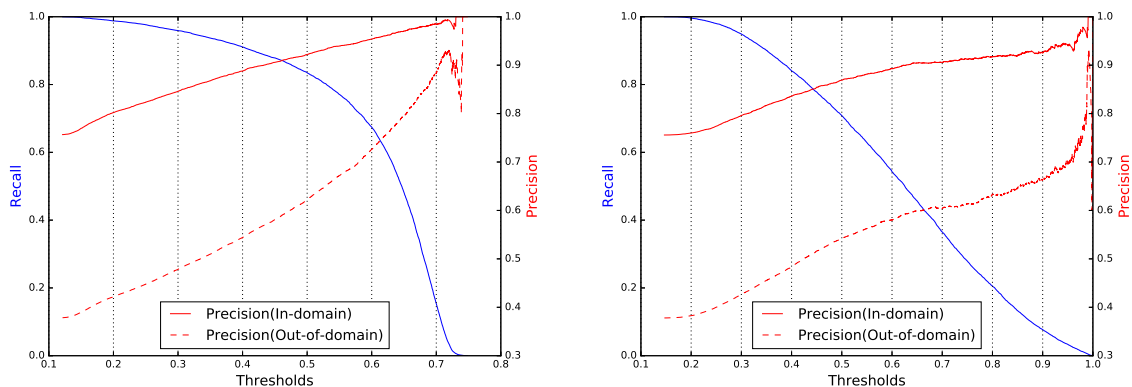
Figure 6: In-domain and out-of-domain performances (precision/recall with respect to filtering threshold) using whitebox GBM meta-model (left) and base model scores (right)
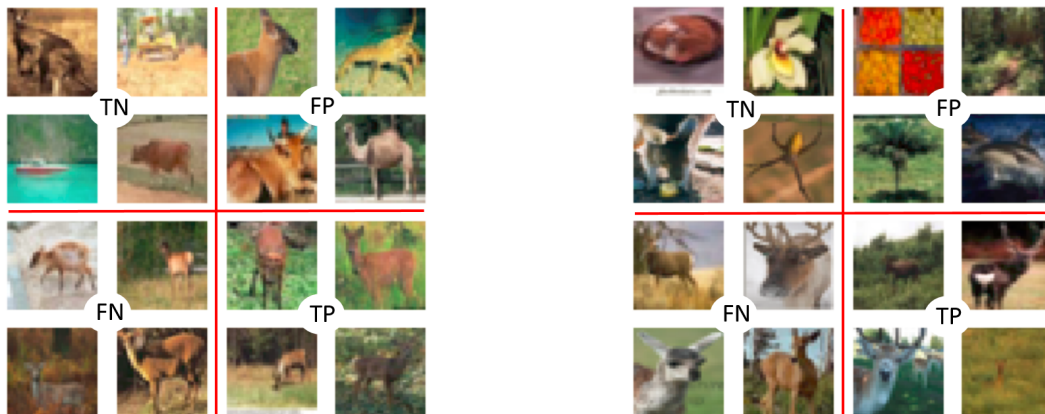

Figure 7: Confusion-quadrant examples for the whitebox logistic regression (left) and the base model score (right). TN: true negatives, FP: false positives, FN: false negatives and TP: true positives.

from the Noisy/Noisy out-of-domain test set.[4] Figure 7 compares image examples sampled from the confusion quadrants when using the meta-model scores (left-hand side) with those sampled using the base model class score (baseline, right-hand side). The thresholds for each system were chosen so as to achieve highest precision while still obtaining at least four samples in each confusion quadrant. Representative images shown in Figure 7 were randomly sampled from the resulting quadrant sets. Subjectively, it appears that the FP images from the whitebox meta-model are relatively competitive with the "deer" class compared to ones which the simple baseline falsely accepts. A similar, albeit subjective, assessment in favor of the meta-model can be made comparing the FN images across the two systems.

---

[4]An interesting article showing some CIFAR examples of false positives can be found at `https://hjweide.github.io/quantifying-uncertainty-in-neural-networks`.

7   Conclusion and future work

We proposed the paradigm of meta-models for confidence scoring, and investigated a whitebox meta-model with linear classifier probes. Experiments on CIFAR-10 and CIFAR-100 data showed that our proposed method is capable of more accurately rejecting samples with low confidence compared to various baselines in noisy settings and/or out-of-domain scenarios. Its superiority over blackbox baselines supports the use of whitebox models and our results demonstrate that probes into the intermediate states of a neural network provide useful signal for confidence scoring.

Future work includes incorporating other base model features. One example is the work by (Gal et al., 2017) whereby the uncertainty measures using Monte Carlo dropout could serve as additional features to our proposed whitebox meta-model.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proc. KDD*, pages 1721–1730. ACM.

C. Ferri, J. Hernández-Orallo, and R. Modroiu. 2009. An experimental comparison of performance measures for classification. *Pattern Recogn. Lett.*, 30(1):27–38.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Proc. ICML*.

Yarin Gal, Jiri Hron, and Alex Kendall. 2017. Concrete dropout. *arXiv preprint arXiv:1705.07832*.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Proc. NeurIPS*, pages 4885–4894.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proc. ICML*, pages 1321–1330.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016a. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Identity mappings in deep residual networks. In *Proc. ECCV*, pages 630–645.

ICML Workshop. 2006. *Third Workshop on ROC Analysis in ML*, ICML Workshop.

Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. 2011. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274.

Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. 2017. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*.

Michael Kampffmeyer, Arnt-Borre Salberg, and Robert Jenssen. 2016. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

Alex Kendall and Roberto Cipolla. 2016. Modeling uncertainty in deep learning for camera relocalization. In *Proceedings of the 2016 IEEE international conference on robotics and automation (ICRA)*, pages 4762–4769. IEEE.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Proc. NeurIPS*.

Christian Leibig, Vaneeda Vaneeda Allken, Murat Seckin Ayhan, Philipp Berens, and Siegfried Wahl Wahl. 2017. Leveraging uncertainty information from deep neural networks for disease detection. *bioRxiv doi: 10.1101/084210*.

National Highway Traffic Safety Administration. 2017. PE 16-007. Technical report.

J. Navrátil and G.N. Ramaswamy. 2002. DETAC - a discriminative criterion for speaker verification. In *Seventh International Conference on Spoken Language Processing (ICSLP)*, Denver, CO.

Sheng Wang, Siqi Sun, and Jinbo Xu. 2016. Aucmaximized deep convolutional neural fields for protein sequence labeling. In *Proc. ECML PKDD*, pages 1–16.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proc. ICML*, pages 609–616, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proc. KDD*, pages 694–699. ACM.

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

Kelly H Zou, Aiyi Liu, Andriy I Bandos, Lucila Ohno-Machado, and Howard E Rockette. 2011. *Statistical evaluation of diagnostic performance: topics in ROC analysis*. CRC Press.