CONFIDENCE SCORING USING WHITEBOX META-MODELS WITH LINEAR CLASSIFIER PROBES

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a confidence scoring mechanism for multi-layer neural networks based on a paradigm of a base model and a meta-model. The confidence score is learned by the meta-model using features derived from the base model–a deep neural network considered a whitebox. As features, we investigate linear classifier probes inserted between the various layers of the base model and trained using each layer's intermediate activations. Experiments show that this approach outperforms various 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 exploring various aspects of the method.

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 can 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 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 other (fall-back) sensors, or choose a conservative action of slowing down the vehicle (Kendall & Gal (2017)). Lack of, or poor confidence estimates can possibly result in loss of human life (NHTSA (2017)).

In this paper we address this problem by pursuing the following paradigm: a learnable confidence scoring mechanism acts as an "observer" (*meta-model*) on top of an existing neural classification model (*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, we would like to have a meta-model G that, given a base model $y = F(x)$, produces a confidence score $z = G(x, \mathbf{\Theta}_F)$ (where $\mathbf{\Theta}_F$ denotes the parameters of the base model). 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 & Bengio (2016)) inserted into the intermediate layers of the base model (hence referred to as whitebox due to its transparency with respect to the internal states). We use a well-studied task of 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, especially when noisy data are considered in the training. By removing samples deemed most uncertain by our method, the precision of the base model 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 capable of rejecting these with better accuracy than baselines.

2 Related work

The work on Monte Carlo dropout (Gal et al. (2017), Gal & Ghahramani (2015)) to estimate model uncertainty can be applied to the filtering task at hand. In an autonomous driving application this approach showed that model uncertainty correlated with positional error (Kendall et al. (2016)). In an application to image segmentation uncertainty analysis was done at the pixel level and overall classification accuracy was improved when pixels with higher uncertainty were dropped (Kampffmeyer et al. (2016)). Monte Carlo dropout was used to estimate uncertainty in diagnosing diabetic retinopathy from fundus images (Leibig et al. (2017)). Significant diagnostic performance improvement was reported when uncertainty was used to filter out some instances from model based classification.

Uncertainty estimates from methods like Monte Carlo dropout can be viewed as providing intrinsic features about a model's prediction for an instance, which can be subsumed by the meta-model approach we are proposing.

In a broader context, the ability to rank samples is a fundamental notion in the Receiver Operating Characteristics (ROC) analysis. The 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 across a variety of disciplines, including biomedical, signal, speech, language, and image processing, has been done in the context of filtering and decision making (Zou (2011), ROC (2006)). Moreover, the 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 & Ramaswamy (2002)). Because 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.

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. Guo et al. (2017) proposed calibration, a form of confidence scoring, as a postprocessing step to mitigate the problem of miscalibration, rendering neural models more interpretable. Due to the fact that Guo et al. (2017) performs calibration after the result of the base model by fitting a stepwise monotonic function (e.g., histogram binning (Zadrozny & Elkan (2001)), or isotonic regression (Zadrozny & Elkan (2002))), this step does not alter the ranking of confident vs. uncertain samples and has therefore have no relevance in our setup.

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 (*MM*, the *meta-model*) operating on F (*base model*) and producing a 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 $\hat{\mathbf{y}}$:

$$z = MM_{blackbox}(\hat{\mathbf{y}}). \tag{1}$$

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 just taking the probability output of the predicted class label:

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

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 probability), and $\mathbf{\Theta}_F$ denotes the parameters of the base model F.

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

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

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

$$z = MM_{whitebox}(\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n), \tag{4}$$

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 & 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 degrees of abstractions of the input data: lower layers (those close to the input) learn more elementary patterns whereas higher layers (those close 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:

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

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:

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

This architecture is illustrated in Figure 1. The diode "—▷|—" symbol is used to direct the information flow one way, and to emphasize that the probes are not trained jointly while training the base model, instead they are trained with the underlying base model fixed.

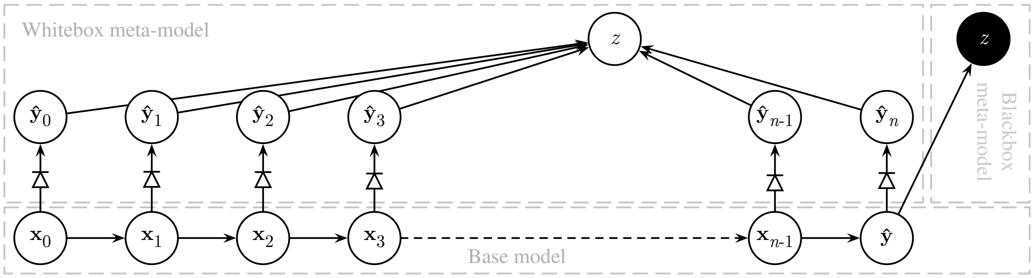


Figure 1: Base model vs. whitebox & blackbox meta-models.

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 $P(1|\hat{\mathbf{y}}_1, \cdots, \hat{\mathbf{y}}_n, \mathbf{\Theta}_G)$ is used as the confidence score z.

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

$$z = \frac{e^s}{1 + e^s} \text{ with } s = \boldsymbol{\theta} \cdot \begin{bmatrix} \hat{\mathbf{y}}_1 & \hat{\mathbf{y}}_2 & \cdots & \hat{\mathbf{y}}_n \end{bmatrix} . \tag{7}$$

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.

2-layer neural network (NN) The concatenated linear probe vectors are passed through a 2-layer neural network with sigmoid activations (sigmoid in the final layer, since its output has 2 dimensions).

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

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: the in-domain confidence scoring task and 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.

Out-of-domain task Given a base model (here again trained on CIFAR-10, hence would be able to classify images into 10 classes), what would the model do if presented with images not belonging to these 10 classes? The predictions made by the base model will surely be wrong: However, can the meta-model deduce that these predictions are 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. 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 data partition	New data partition	Size	Usage
Original 50,000 train	TRAIN-BASE	30,000	Training the base model
	TRAIN-META	10,000	Training the meta-model
	DEV	10,000	Tuning both models
Original 10,000 test	TEST	10,000	Held-out test set

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.

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

- 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.

4.2 BASE MODEL

We reuse the state-of-the-art 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 (He et al. (2016a;b); Zagoruyko & Komodakis (2016)) of convolution networks 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 $\hat{\mathbf{y}}$.

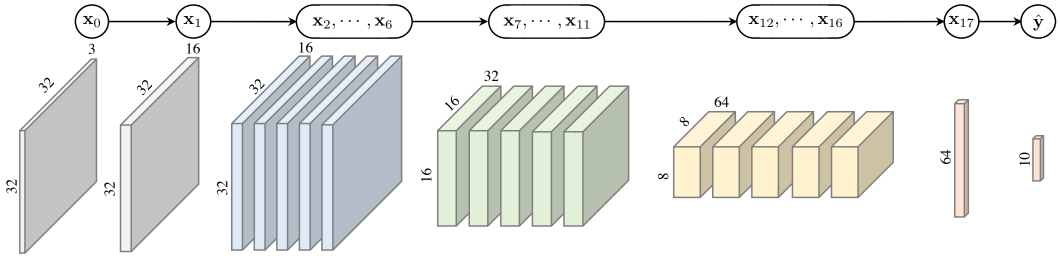

$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2, \cdots, \mathbf{x}_6$
$\mathbf{x}_7, \cdots, \mathbf{x}_{11}$
$\mathbf{x}_{12}, \cdots, \mathbf{x}_{16}$
$\mathbf{x}_{17}$
$\hat{\mathbf{y}}$
3
32
32
16
32
32
16
32
32
32
16
16
64
8
8
64
10

Figure 2: Neural structure of the base model.

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 as drawn from the CIFAR-10 dataset;

Noisy base / Clean meta The training set TRAIN-BASE is 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). We consider this scenario a proxy for tasks that are inherently harder and have higher error rates than the relatively accurate original base model on CIFAR-10.

Noisy base / Noisy meta In this case the sets TRAIN-META and DEV, too, are corrupted by same label noise (30% of samples) as above. This condition, in combination with the degraded base model, represents a realistic scenario of obtaining training data from a noisy environment, e.g., via crowd-sourcing in which labels are not always correct.

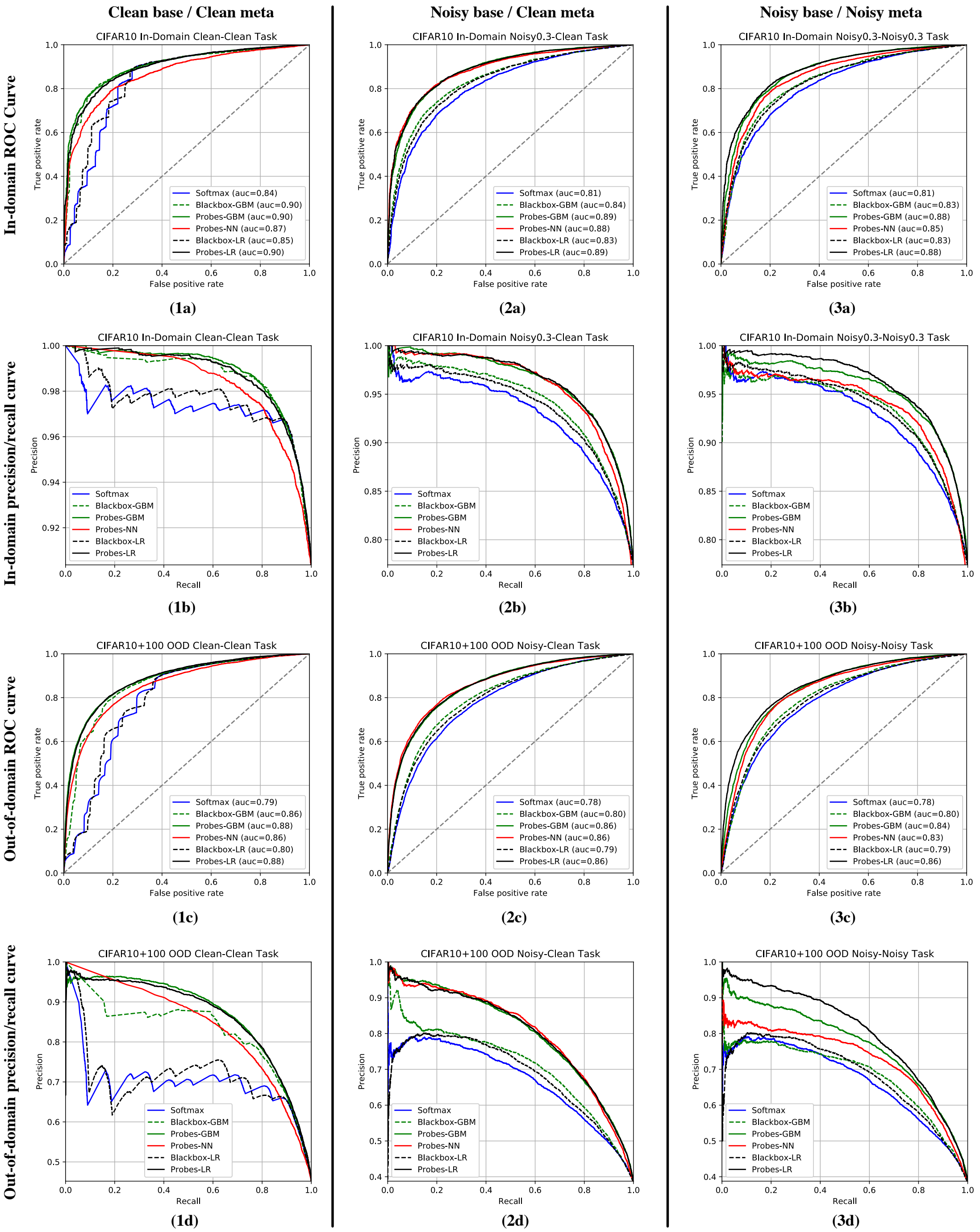
These conditions in combination along with the two tasks offer a representative spectrum of classification scenarios encountered in practice. It should be pointed out that in all conditions the TEST set (both *in-domain* and *out-of-domain*) is applied clean without artificial corruption.

We compare the following methods for confidence scoring: (**Softmax**) Simple blackbox softmax using Eq. (2); (**Blackbox-LR/GBM**) Use the final output $\hat{\mathbf{y}}$ as the only feature for the meta-models; (**Whitebox-LR/NN/GBM**) Use 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.

The results are listed below.



(1a) (2a) (3a)

(1b) (2b) (3b)

(1c) (2c) (3c)

(1d) (2d) (3d)

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. Similar plots for the 30%-degraded version of the base model but clean meta-models are shown in Figures (2a)-(2d). Finally, 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 (3a)-(3d).

6 Discussion

The experimental results presented in the earlier 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 three scenarios: clean base / clean meta, noisy base / clean meta, noisy base / noisy meta, respectively. The impact of noise is seen in the top accuracies achieved in two of the three scenarios. The accuracies improve with depth for the most part in all three scenarios.

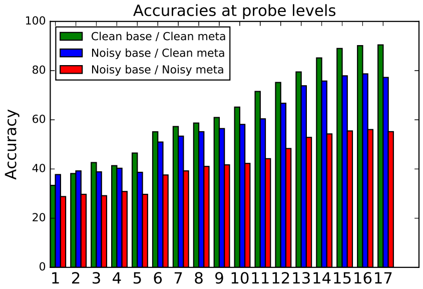


Figure 4: Accuracies for the meta-model training data at varying probe levels

The accuracy plots do not provide insights into how the whitebox models achieve their higher performances and 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)). Considering the clean data scenario first, the top features based on their relative importance estimation for GBM models include the probe outputs at the deepest layers (17, 16, 15, 14) corresponding to the predicted class with the highest final base model score and the class with the second highest base model score. This aligns with the intuition that having high scores for the predicted class and large gaps between the scores for the top two classes might be indicative of the base model being correct. This changes when we consider the two scenarios with label noise. The most important features for the GBM whitebox model in these cases are the probe scores for the predicted class at intermediate layers (11, 12, 13) followed by the second last layer (15). The ability of the meta-model to utilize the information from the intermediate probes leads to its significant improvement in performance in the two noisy scenarios.

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. Consider the scenario where label noise is added to data for both base and meta-models. One could argue that this scenario is the most relevant for many real-life applications where labels in training data can be quite noisy. Figure 5 shows the comparative performances in in-domain and out-of-domain settings for the whitebox logistic regression meta-model and the base model final scores, respectively.

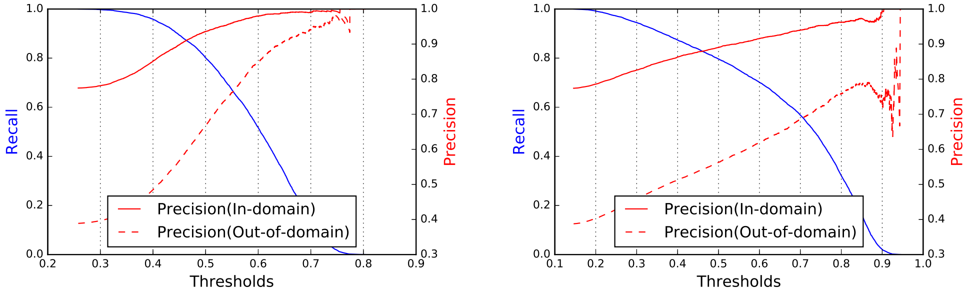


Figure 5: In-domain and out-of-domain performances using whitebox logistic regression meta-model (left) and base model scores (right)

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 5 (left). Let's say, in an application setting, we pick a threshold (≈0.5) that achieves an in-domain recall of 0.8. At this threshold, the logistic regression whitebox meta-model achieves an in-domain precision greater than 0.9. If we encounter a domain shift as represented by the out-of-domain task the precision degrades to ≈0.65. Consider the same situation if we were using the base model score as in Figure 5(right). The in-domain precision is similar (≈0.9) but the drop in precision for the out-of-domain case is steeper to ≈0.55. 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.

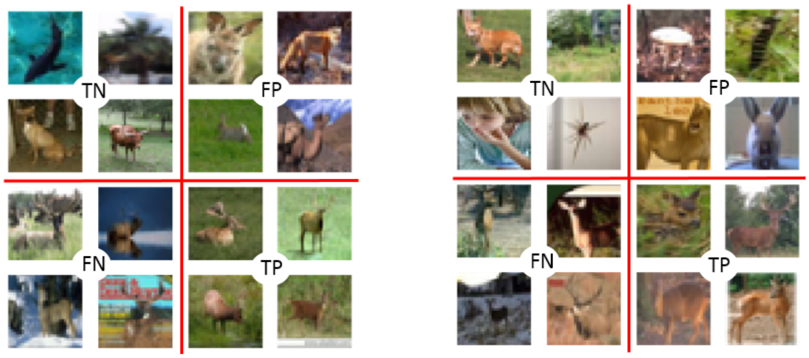


Figure 6: Confusion-quadrant examples for the whitebox logistic regression (left) and the base model score (right).

The impact of meta-model based filtering can be further illustrated using examples representing four quadrants of the binary confusion matrix (TN - true negatives, FP - false positives, FN - false negatives and TP - true positives). We chose the CIFAR-10 class "deer" and considered all instances from the out-of-domain test set[4]. Figure 6 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 6 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.

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, especially 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 MC dropout could serve as additional features to the whitebox meta-model.

References

Third Workshop on ROC Analysis in ML, ICML Workshop, Pittsburgh, PA, USA, June 29 2006.

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

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

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

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

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1721–1730. ACM, 2015.

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

Yal Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *arXiv preprint arXiv:1506.02142*, 2015.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/guo17a.html`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645. Springer, 2016b.

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

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

Michael Kampffmeyer, Arnt-Borre Salberg, and Robert Jenssen. 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, June 2016.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30*, 2017.

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

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

J. Navrátil and G.N. Ramaswamy. DETAC - a discriminative criterion for speaker verification. Denver, CO, September 2002.

NHTSA. Pe 16-007. Technical report, 2017.

Sheng Wang, Siqi Sun, and Jinbo Xu. *AUC-Maximized Deep Convolutional Neural Fields for Protein Sequence Labeling*, pp. 1–16. Springer International Publishing, Cham, 2016.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 609–616, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL `http://dl.acm.org/citation.cfm?id=645530.655658`.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699. ACM, 2002.

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

K.H et al. Zou. *Statistical Evaluation of Diagnostic Performance - Topics in ROC Analysis*. CRC Biostatistics Series. CRC Press, 2011.