Scoring Confidence in Neural Networks

Nikita Vemuri

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2020-132
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-132.html

June 2, 2020

Scoring Confidence in Neural Networks

by Nikita Vemuri

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Joseph Gonzalez
Research Advisor

May 28, 2020

(Date)

* * * * * * *

Professor Ion Stoica
Second Reader

May 27, 2020

(Date)

Scoring Confidence in Neural Networks

by

Nikita Vemuri

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Joseph Gonzalez,
Professor Ion Stoica

Spring 2020

Abstract

Scoring Confidence in Neural Networks

by

Nikita Vemuri

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Joseph Gonzalez

Increased trust in the predictions of neural networks are required for these models to gain greater popularity in real world decision making systems where the cost of misclassification is high. Accurate confidence estimates which represent the expected sample accuracy can greatly aid in increasing this trust, however modern neural networks are largely miscalibrated. Confidence estimates can be used as interpretable probabilities which are fed into the next stage of the decision making system, or as values which determine when not to act on the neural network's predictions when compared to a threshold. We explore the effects of various regularization techniques and calibration methods on the expected calibration error. We find that widely regularization techniques do decrease the calibration error, but that the best hyperparameter values for this regularization may be different than the value that maximizes the generalization accuracy. For the application of using confidence estimates to determine which inputs to not predict on, we develop a framework of visualizing the tradeoff between the proportion of inputs not predicted on and the resulting accuracy. This method has a greater flexibility in the types of confidence scores that result in good indicators of when to trust the model and can also support confidence scores that are not probabilities or direct estimates of the expected sample accuracy. We demonstrate this using entropy and distance to the decision boundary as two methods which can separate out points that are likely to be incorrect without returning probabilities that are interpreted as traditional confidence estimates.

Contents

List of Figures

Acknowledgments

I would like to thank my advisor Professor Joseph Gonzalez for his continual support and detailed guidance over the last three years. I am truly grateful to have spent time in the RISE Lab, which sparked my interest in the intersection of machine learning and systems and shaped my goals for the future. I would also like to thank Soeren Kuenzel and Professor Jasjeet Sekhon for introducing me to the world of research and serving as mentors and friends over the years. Finally I would like to thank my friends and family, who were my strongest support system and constantly motivated me to do my best.

Chapter 1

Introduction

As neural networks are able to obtain higher accuracies across a variety of problems, there is an increased demand to deploy them in applications where the cost of misclassification is high, such as medical decision making systems [1]. However, in order for neural networks to gain widespread popularity in these domains, there needs to be a more reliable notion of predictability and trust in the performance at test time. Many of these high risk applications have long established legacy processes that can perform the task, such as having a human expert make a classification. Although these processes are generally more trusted, they are very costly to use at large scales.

For classification problems, a critical aspect to maintaining trust in a model's performance is developing estimates in the confidence of the prediction that reflect the true expected accuracy of that sample. This would allow a practitioner to not only better understand the chance of the model predicting incorrectly on a per sample basis, but also potentially use that estimate to determine when to default to the legacy process.

There are two main uses for estimates of the confidence of a prediction. Some applications directly need the confidence estimate as an output of the model, which is then used in the next step of the decision making process. For these applications, it is especially important the confidence estimate represents the expected sample accuracy, so the natural interpretation of the probability is valid. For example, a doctor may want to use the probability that a patient has a disease in combination with other information to make the final diagnosis, so the probability must have the expected meaning. In other applications, a threshold can be set on the confidence estimate to determine when the neural network should not make a prediction. Recent work has shown that modern neural networks are overconfident in their predictions despite a higher generalization accuracy, thereby highlighting the need for calibration methods [6]. In addition, it has been shown that not all points in a dataset are equally difficult to classify [21].

We address the two main applications of confidence estimates in this work. Guo et al. [6]

indicates that a key reason behind the trend of increasing miscalibration is that modern neural networks can overfit to the negative log likelihood loss without overfitting to the 0/1 loss that is used to determine generalization accuracy. As a result, we first investigate the role of widely used regularization techniques on the calibration error.

We then look at various methods of calibrating outputs of neural networks, such as histogram binning, matrix scaling, and temperature scaling, which can provide a better direct estimate of the expected sample accuracy. These methods are well suited to analysis using reliability diagrams [15], which visualize the calibration error by plotting the confidence estimate against the expected sample accuracy .

In addition, we develop a method of directly analyzing the tradeoff between only using the neural network's predictions on a subset of the data (ignoring predictions that are assigned low confidence estimates based on some threshold) and the resulting accuracy of the model. This framework is simple to use on the validation dataset for any classification problem, and is well suited for the case when inputs arrive in a stream during deployment. It also allows for the usage of confidence scores, which are values that may not represent the expected sample accuracy but still give some indication of the confidence in the prediction. Because they don't represent the probability of correct prediction, confidence scores should not be used in the same way as a confidence estimate, even if the values are probabilities.

This framework is especially useful in applications which have a robust and tested legacy process to perform classifications, and are shifting to using neural networks for the task. These types of applications tend to have a high cost for misclassification, so predictability of the model's performance is very important. Only predicting on the inputs where the neural network produces confident predictions allows for a gradual transition to automate the legacy process, while still maintaining a high level of trust. It also gives the practitioner the ability to tune the expected accuracy of the system by choosing the proportion of inputs to defer to the expert.

Using this framework, we analyze methods that provide indications of the likelihood of correctness without directly calibrating to the expected sample accuracy. For example, we use approximations of the distance to the decision boundary as proposed in Elsayed et al. [3] as scores that may indicate the likelihood of correct prediction and find performance that is almost equal to temperature scaling. Entropy of the predicted distribution over class labels is also shown as a reasonable confidence score, even though it doesn't correspond to the sample accuracy. These alternative scoring methods are not required to be probabilities, so are not well suited for reliability diagrams. However, our method of visualizing the best accuracy for various sized subsets allows a framework that can compare confidence scoring methods regardless of whether they have the traditional intepretation of a confidence estimate.

Chapter 2

Related Work

Uses of Accurate Confidence Estimates

As machine learning models get deployed in real world decision making systems where the cost of a misclassification is high, well calibrated confidence estimates become increasingly important. A confidence estimate can be considered well calibrated if the estimate is sufficiently close to the probability of that input being correctly classified. In practice, we can obtain an estimate of the probability of correct classification by taking the sample average accuracy of all data points with the same features. In situations where there are not a sufficient number of data points with identical features, a grouping can be done on similar inputs.

Certain applications use the confidence estimate from a discriminative model as an input to the next level of the decision making process. Jiang et al. [10] describes a study in which the confidence estimates from ICU mortality calculators are used to determine whether or not to discontinue various types of therapy. In these types of applications, it is important the confidence estimate is a probability that takes on the same intuitive meaning humans would expect out of a probability, as the next step is usually determined on that assumption. Rather than simply comparing the confidence estimate of the predicted class to a threshold, the entire estimated probability distribution across all possible classes can be used as an input to another model. An interpretable probability estimate is especially required if human expert will make a recommendation on that value.

On the other hand, confidence estimates in combination with a threshold can be used to obtain an indicator of whether to trust the model's predictions. This can seamlessly be used in the example of automated medical diagnoses, as the model can rely on an expert for inputs which cannot be predicted with sufficient confidence [10]. Because the legacy process of receiving a diagnosis from a doctor can be treated as the expert's prediction and there is a high cost for incorrect diagnoses, the model should only be used when the user can trust that the its prediction will be correct. In this case, it is not necessary that the confidence

estimate is interpretable as a stand-alone quantity, but instead that it can be used to develop a good indicator of trust in the model predictions.

Calibration of Modern Neural Networks

For classification problems, applying a softmax layer on the output of the neural network gives a natural probability distribution over the output space that can be used as a confidence estimate. However, recent work has shown that unlike some of their predecessors, modern neural networks are poorly calibrated even though they tend to have higher generalization accuracies [6]. The authors study a variety of changes to neural network design and training in recent years, and ultimately correlate this trend with increases in model capacity and a form of overfitting.

Although the test error does decrease throughout the course of training a large ResNet model on CIFAR-100, the authors find that the negative log likelihood (NLL) increases after a certain point, indicating that the model overfits on the NLL loss without overfitting on the test accuracy [6]. This is especially possible with the NLL loss because even after all the train points are correctly classified, the loss can be further decreased by pushing the estimated probability distribution across output classes to be closer to a soft-indicator function. As a result, the probability estimates from modern neural networks can be overconfident. These results are in line with Zhang et al. [24], which state that deep neural networks can counter the traditionally supported notion that large models generalize poorly without regularization. Relating to these findings, the authors of Guo et al. [6] suggest the overfitting that is observed during training doesn't show up in the generalization error, but instead in the accuracy of the confidence estimates. Because miscalibration of confidence estimates can be attributed to overfitting, in this work we will explore the calibration error when various forms of regularization are used.

Prior work has explored calibrating the confidence estimate using an ensemble of neural networks, however, unless the task to solve already requires an ensemble model, training multiple networks simply for the purpose of calibration becomes expensive [12]. Other work has investigated getting an estimate of the uncertainty of a prediction by using information collected from stochastic forward passes of a network trained with dropout, which also makes certain assumptions on the model architecture [4].

Alternate Approaches to Quantifying Model Uncertainty

Rather than directly recalibrating the confidence estimates, models can be trained with an emphasis on better estimating the uncertainty in a prediction. Gao et al. [5] present a solution for the problem of ambiguous labels by training neural networks on the observed label distribution rather than on a one-hot vector. This is similar to the idea of label smoothing which is used to regularize neural networks, but takes into account the true

labels of inputs in the training set and accordingly defines the level of smoothing. Seo, Seo, and Han [17] introduce a loss function that is a combination of the cross entropy loss with respect to the true label distribution and the cross entropy loss with respect to the uniform distribution across labels. These two terms are weighted by the variance of the predictions from stochastic inferences of an input, and networks trained with this method have predicted confidence estimates that are better calibrated.

Mixup is another technique introduced primarily to increase the generalization abilities of neural networks through regularization, but also has been shown to increase the robustness of models and the overall calibration of confidences [25]. It is performed by augmenting the training data with inputs that are a convex combination of both the features and labels of two training examples in the original dataset. Although initially developed to help neural networks generalize better, Thulasidasan et al. [19] show that mixup can improve the calibration of neural networks. The benefit from mixup on calibration is is greatly dependent on the label smoothing aspect of the procedure that is not present when only data augmentation is done using a combination of the input features.

The training procedure of neural networks can be modified with various techniques that better the model calibration, but this work primarily focuses on procedures that occur after the training of a model that can further boost the calibration of the confidence estimates.

If the goal of calibrating uncertainty estimates is to get a better understanding and finer grained control of how the neural network may perform at test time, methods can be used to directly accomplish the desired task, rather than first calibrating the confidences. Trappenberg and Back [20] uses an "I Don't Know" (IDK) class to highlight areas of the input space with high uncertainty, and accordingly relabel the training data so the model performs better. Wang et al. [21] reduce computational cost of a prediction without decreasing the accuracy by cascading the input to a more expensive model with higher accuracy if the cheaper initial model predicts the IDK class. A particularly interesting result from this work is that some inputs are "inherently easier" than others, as quantified by multiple models of varying complexity and accuracy classifying those inputs correctly [21]. This indicates there is some natural separation between easy and hard inputs, that a good confidence calibration method should be able to pick up on.

One may also care about accurate confidence estimates from the perspective of predicting the accuracy of a model at test time. Jiang et al. [11] directly predicts the generalization gap of a model by regressing a signature of the distribution of margins. The margin can be defined as the minimum distance to the decision boundary from the data points, and is often used in context of SVMs. The margin cannot be analytically calculated for neural networks, unlike for more classical models, so Elsayed et al. [3] propose an approximation of the margin under a l_p norm.

Directly analyzing the quantity that best defines model uncertainty for a given problem may work well as an alternative to calibrating confidence estimates to use them for another task. We take inspiration from these various techniques and strategies to eventually determine the exact tradeoff between accuracy and deferring a greater proportion of inputs to the expert.

Chapter 3

Framework for Quantifying Calibration

The calibration of a model can be quantified using reliability diagrams, which visualize the relationship between confidence and accuracy [2] [15] [6]. Throughout this work, we assume a classification problem where the the training and testing data comes from the same underlying distribution. In the multiclass classification problem, let the data be (x, y) where $y \in \{1 \dots K\}$. Let the neural network be defined as h, where $h(x, y) = (\hat{y}, \hat{p})$. $\hat{y}$ is the network's prediction of the class label and $\hat{p}$ is the confidence value associated with that prediction, usually obtained from applying the softmax function on the network's outputs. A perfectly calibrated model would produce confidence values which exactly match the expected sample accuracy. In other words, if a perfectly calibrated model produces multiple predictions with confidence of $\hat{p}$ then we would expect that a $\hat{p}$ fraction of those predictions are correct.

However, in practice it is highly unlikely that there will be sufficient identical inputs to the model to get a good estimate of the expected sample accuracy, so instead, the model predictions are grouped into M bins of equal length based on their confidence values. The accuracy of bin B_m is given by

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i)$$

The average confidence in a bin B_m is given by

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

The reliability diagram is generated by creating a bin B_m of height $\text{acc}(B_m)$ for each $m \in \{1 \dots M\}$. A bin B_m would be considered perfectly calibrated if $\text{acc}(B_m) = \text{conf}(B_m)$.

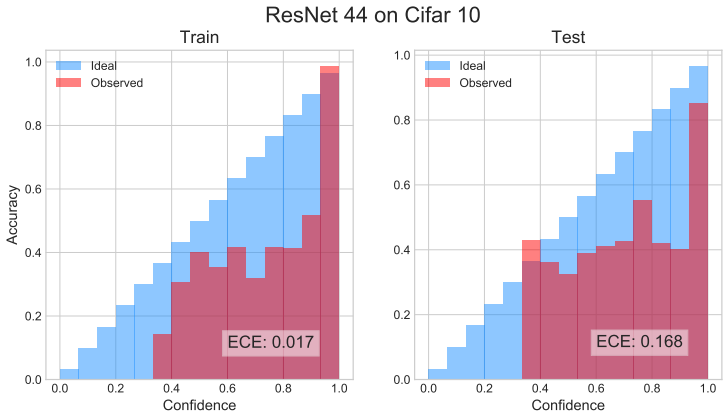


Figure 3.1: Reliability diagrams for a 44-layer ResNet on CIFAR-10 shows that modern neural networks are overly confident in their confidence estimates.

Reliability diagrams can give information about the miscalibration within each bin, but don't indicate the miscalibration across all predictions as the number of samples per bin is not displayed.

The Expected Calibration Error (ECE) [14] is a summary statistic of the calibration across all the predictions, and is obtained by weighting the difference between the accuracy and confidence of each bin with the number of samples in that bin.

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|$$

Miscalibration of Confidence

Throughout the years, modern neural networks have demonstrated an improved accuracy across a variety of tasks, but Guo et al. [6] showed that these gains in accuracy come at the cost of miscalibrated confidence outputs. Figure 3.1 replicates this result on a 44-layer ResNet [7] on the CIFAR-10 dataset for both the training and testing data. While the predictions on both datasets are overconfident, the test set shows a much higher ECE.

Guo et al. [6] analyzes various potential causes of increasing miscalibration in current neural networks and primarily links greater model capacity and a lack of sufficient regularization to this trend. They show that given current practices of model architectures and optimization techniques, neural networks can overfit to the negative log likelihood (NLL) loss function without overfitting to the 0/1 loss.

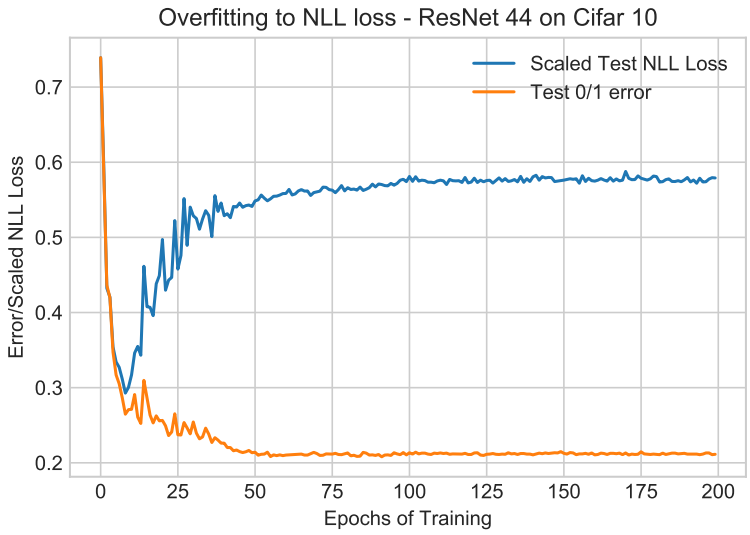


Figure 3.2: Modern neural networks can overfit to the NLL loss without overfitting to the 0/1 error.

This results in models that generalize well to the test set in terms of accuracy, even though the NLL loss on the test set may increase after a certain amount of training, as replicated in Figure 3.2. Although the overfitting to the NLL loss is not present in the classification error, Guo et al. [6] state that this overfitting manifests itself in miscalibration of confidence estimates. This effect is possible when models are trained with the NLL loss because even after all of the training points are classified correctly, the loss can be further decreased by pushing the confidence estimates of the predicted class towards one.

Chapter 4

Framework for Flagging Likely Incorrect Points

Meaningful confidence estimates can increase the trust in a neural network's performance when it is deployed in a real world decision making system. Neural networks may be introduced to fully or partially replace a legacy system that is reliable yet expensive to use at scale. In these scenarios, calibrated confidence estimates can be used to guide a practitioner on when to let the system act on the prediction of a neural network and when to revert to the legacy system. The use for the confidence estimates then becomes to separate out points that are likely incorrect from those that are likely correct.

This idea has been explored through the use of "I Don't Know" (IDK) uncertainty classes [20] [21], which the model can predict if there is not sufficient confidence in any of the original classes. Confidence can be defined as either the softmax probability for the predicted class or some transformation on the output of the neural network. Once some threshold is crossed, the model predicts the IDK class. For example, Wang et al. [21] explore various methods as the confidence values, including using a linear transformation and the entropy on the model outputs, and determine a threshold for the IDK class through grid search. Although the threshold for the IDK class is crucial for the final deployment process, a practitioner may be more interested in an estimate of how much data the model will defer to a oracle at test time and the resulting accuracy, rather than the value of the threshold.

We present uncertainty class threshold diagrams as a direct visualization of this process. These diagrams plot the accuracy when predictions are made on various subsets of data that are generated from a range of thresholds on the confidence score. If we assume scores below a threshold are considered uncertain predictions, each point on the diagram represents a confidence score that is used as a threshold. The proportion of predictions with a greater score than that threshold and the accuracy of the resulting subset are accordingly plotted. Figure 4.1 shows these curves for the reliability diagram in fig. 3.1 as well as in the ideal

case.

When determining the maximum proportion of the dataset that the neural network can predict on while still maintaining some desired accuracy, we want a measure of confidence that reliably flags likely incorrect points by assigning them low scores. This is in contrast to the task of perfect confidence calibrations where the goal is to obtain a confidence estimate equal to a specific value. The ideal curve for this plot is accordingly created by setting the confidence values to be equal to the indicator of whether the point is correctly classified.

$$\text{conf}_{\text{ideal}}(h, x_i) = \mathbb{1}(\hat{y}_i = y_i)$$

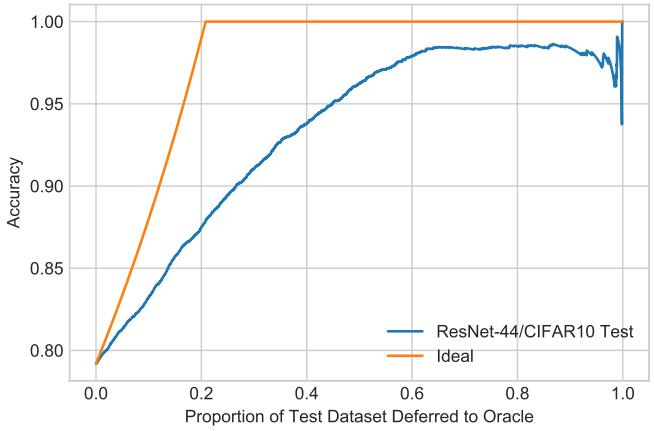


Figure 4.1: Visualization of the uncertainty class threshold diagram for ResNet-44 trained on CIFAR-10, which shows the accuracy obtained when various proportions of the dataset are deferred to the oracle by using the specified confidence estimate as a threshold. The ideal curve is generated by taking the confidence as the indicator of whether that input was correctly classified. The AAC for the ideal case is 0.023 and the AAC for the ResNet-44 model is 0.066.

Based on this definition of the ideal curve, this plot can be summarized with the area above the curve and below the line $y = 1$ (AAC) on the domain of $0 <= x <= 1$. Confidence metrics that result in lower values of AAC are preferred, as they allow for the model to perform at a higher overall accuracy while still predicting on most of the dataset. Baseline accuracies are factored in to the AAC, with higher baseline accuracies resulting in lower AAC values.

The concept of coverage in machine learning informs what proportion of data the model has enough confidence in to make a prediction. The uncertainty class threshold diagram contains

this information as the proportion of data deferred to the expert given a threshold, but also includes information on the accuracy of the subset that is predicted on. Unlike the receiver operating characteristic (ROC) curve, this plot visualizes the expected accuracy for *different* subsets of the original dataset.

Each point on the uncertainty class threshold diagram has an underlying threshold that is associated with a certain proportion of inputs having greater confidence. Although fig. 4.1 visualizes this on the full test dataset, once a desired accuracy and corresponding proportion of data to defer is decided in advance, the threshold on the confidence estimate that resulted in that point on the curve can be also used in the streaming case to determine which inputs to use a model prediction on.

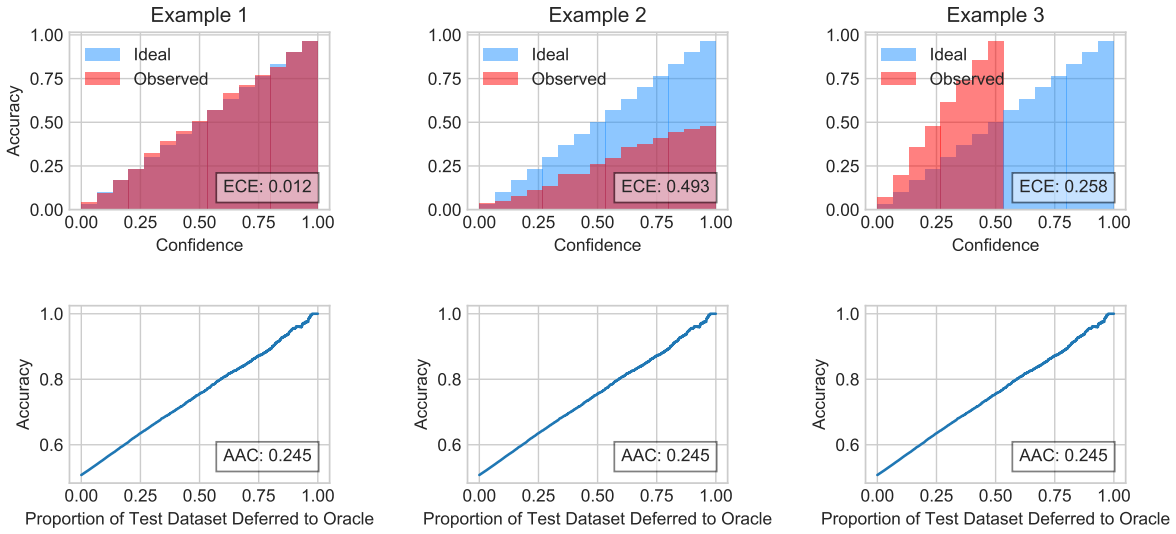


Figure 4.2: The uncertainty class threshold diagrams may allow for more flexibility in a confidence estimation. As shown by these toy examples, multiple confidence estimations for a single problem can result in the same cutoff plot. The first example is generated by setting the probability of a correct prediction be the confidence value exactly. The next two examples show cases where the probability of a correct prediction is a linear transformations on the confidence value.

The uncertainty class threshold diagrams provide a framework that allows calibrating confidence scores to values that don't have to be interpretable as probabilities, because they take into account the relative ordering of confidence scores. By using these diagrams to determine a threshold, the calibration method can be more flexible in its assignment of scores, because

there is no longer a need to precisely estimate the expected accuracy of an input. Figure 4.2 shows a toy example where multiple reliability diagrams all result in the same uncertainty class threshold diagram because the relative ordering is maintained.

Although the uncertainty class threshold diagrams don't provide information on the calibration error and don't even require confidence scores to be probabilities, they are able to visualize how well a particular confidence scoring method can be used to create an indicator of whether to trust the neural network's predictions.

Chapter 5

Effect of Regularization on Calibration

Guo et al. [6] show that although modern neural networks generalize well in terms of accuracy, they still overfit to the negative log likelihood. They suggest that the overfitting doesn't affect the classification error but instead may manifest as miscalibrated confidence estimates. We further evaluate this hypothesis by observing the ECE and test error for various forms of regularization in 5.1.

Label smoothing is a regularization technique in which a neural network is trained using a weighted average of the one-hot encoding of the labels and a uniform distribution as the labels [18]. Recently Müller, Kornblith, and Hinton [13] has shown that in addition to improving generalization accuracy, this technique can be used to decrease the calibration error. Although we don't find label smoothing to produce a large change in the test error, it does affect the ECE more drastically. The label smoothing parameter that results in the best ECE is not the same as the one that results in the best test error, indicating this hyperparameter should be chosen based on whether greater importance is given to the test error or to the ECE at test time.

Dropout prevents co-adaptation and regularizes a neural network by randomly omitting hidden units at training time based on the dropout probability [8]. Although dropping out too many hidden units hurts the test accuracy, large values of dropout seem to decrease the ECE. This may occur because a model with the majority of hidden units dropped out is simpler and has lower model capacity, which limits the amount of overfitting to the NLL that can happen.

Although weight decay is a effective method of regularization for large neural networks, it has been used less frequently in recent times, as other techniques such as batch normalization also act as a regularization mechanism [9]. For a ResNet-44 model trained on CIFAR-10,

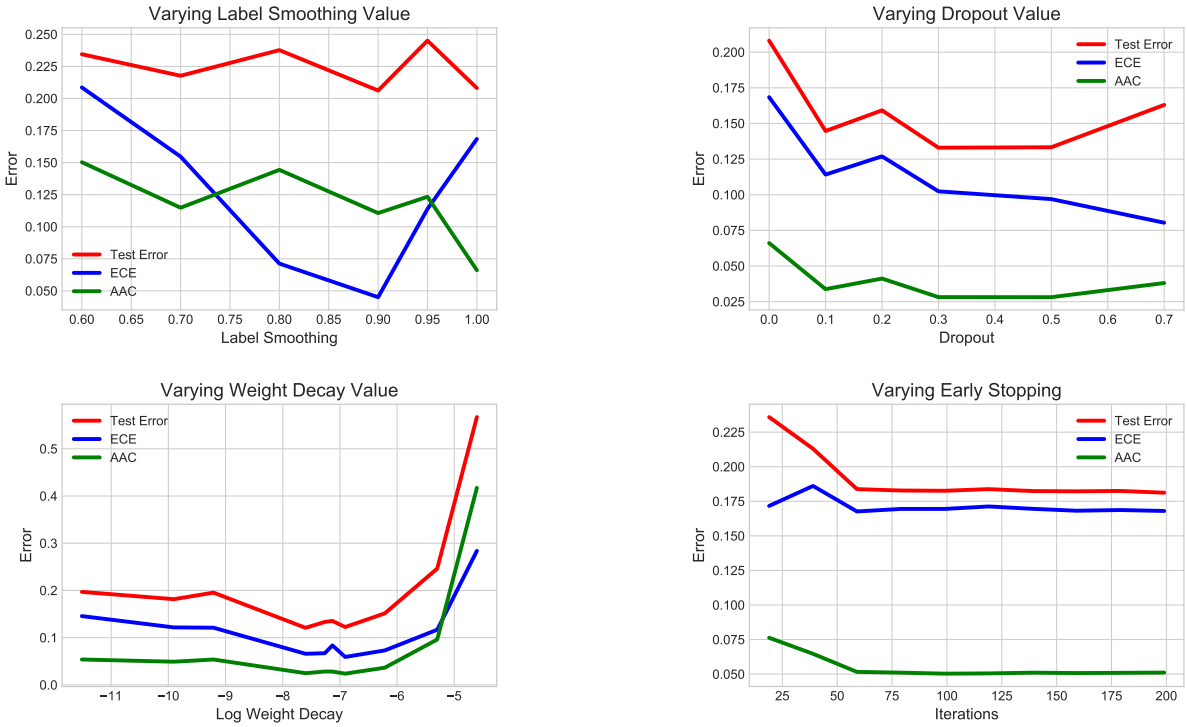


Figure 5.1: Various values for label smoothing, dropout, weight decay, and early stopping were tested to observe the effect on the test error and the ECE. For all of the regularization techniques tested, except early stopping, varying amounts of regularization does affect the ECE for ResNet44 on CIFAR-10. Specifically in the case of label smoothing and dropout, the setting that minimizes the ECE is not the same as the setting that minimizes the test error. This suggests that in applications where accurate calibrations are important, hyper-parameters can be tuned using the ECE as a metric. The AAC of the uncertainty class threshold diagram seems to mirror the test error more closely, partly because higher baseline accuracies result in less space being above the cutoff curve that the confidence metric has to threshold on.

the test error and ECE seem to mirror each other closer than the other previously explored regularization methods. Weight decay does impact the ECE as very small and very large values of weight decay result in higher calibration errors.

Early stopping of training doesn't seem to affect the ECE at test time much differently than the test error. For all of the regularization methods, the AAC for the uncertainty class threshold diagram mirror the test error as it is highly dependent on the accuracy when predicting on the entire test set. In addition to the above regularization methods, we observe that augmenting the training data through standard transforms also decreases the ECE.

Overall, we find that regularizing the neural network at train time does seem to positively impact the calibration error. Moreover, for techniques like label smoothing and dropout, hyperparameters should be optimized with respect to the ECE if that is of higher importance than the generalization error, as the best values under both these metrics can be different.

Effect of Regularization on ECE		
	Train ECE	Test ECE
Standard	0.016	0.168
Data Augmentation	0.006	0.080
Dropout (0.7)	0.008	0.080
Label Smoothing (0.9)	**0.094**	**0.045**
Weight Decay (0.001)	0.005	0.059
Early Stopping (200)	0.016	0.168

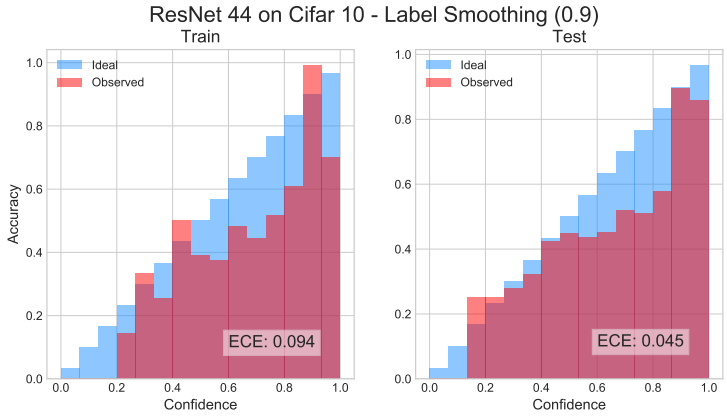


Figure 5.2: Label smoothing with a value of 0.9 works best to reduce the ECE at test time, but as a result makes the model slightly less calibrated at train time.

Chapter 6

Methods of Calibration

In addition to using regularization techniques to get lower calibration errors, one can directly calibrate the confidence estimates from the network. In the following section, we will explore the effect of widely used calibration methods on the ECE at test time and the AAC for the uncertainty class threshold diagrams. Methods that require a learned parameter are trained on the validation set.

Calibration to Estimated Sample Accuracy

Histogram Binning

Histogram binning can be used in multiclass setting to calibrate confidence estimates by treating the problem as K binary classification problems using the one-vs-all method [22] [23]. The following section will use the notation in Guo et al. [6]. For each class label k, let $\hat{p}_i^{(k)}$ be the probability of the input being of class label k. The uncalibrated confidence estimates are then partitioned into M bins $B_1^k \dots B_m^k$ of equal length. For the below experiments, we chose $M = 15$. The calibrated score θ_m^k for bin B_m^k is the average number of samples with label k in that bin. For any input at train or test time, if the uncalibrated probability of class k falls in bin B_m^k, then the calibrated score $\hat{q}_i^{(k)} = \theta_m^k$. Applying this method for the K different classes gives an unnormalized vector of calibrated scores $[\hat{q}_i^{(1)} \dots \hat{q}_i^{(k)}]$ which can then be normalized to get the calibrated confidence values for each class. The predicted class label is given by $\hat{y}_i = \underset{k=1\dots K}{\arg\max}\, \hat{q}_i^{(k)}$.

One limitation of the method is that there are only M possible values of calibrated confidence estimates. If M is increased, one faces the risk of not having enough samples per bin, thereby increasing the variance in the calibrated estimate.

Matrix Scaling

Matrix scaling applies a linear transformation to the logits, z_i, of a neural network and trains the parameters W and b using a negative log likelihood loss function on the validation set. The calibrated confidence then becomes

$$\hat{q}_i = \max_k \sigma(W z_i + b)^{(k)}$$

where σ is the softmax function. The new predicted class label becomes

$$\hat{y}_i = \arg\max_{k=1...K} (W z_i + b)^{(k)}$$

Temperature Scaling

Temperature scaling is an extension of Platt scaling [16] where a scalar parameter $T > 0$ is learned so the calibrated confidence estimate becomes

$$\hat{q}_i = \max_k \sigma(z_i/T)^{(k)}$$

Unlike the other previous methods, scaling by T preserves the order of scores in the original logits vector z_i, so the predicted class label is not changed. T is learned using the negative log likelihood loss function on the validation set.

Calibration to General Confidence Score

Entropy

Entropy can be used to calibrate confidence estimates by directly measuring the uncertainity in the class label given the estimated probability distribution from the output of the softmax layer in the neural network. The entropy function is given as

$$H(x_i) = -\sum_{k=1}^{K} \hat{p}_i^{(k)} * \log_K(\hat{p}_i^{(k)})$$

Higher values of entropy indicate more uncertainty and correspond to a predicted confidence distribution that is closer to uniform. To follow the convention of the previous calibration methods, we define the calibrated score to be

$$\hat{q}_i = 1 - H(x_i)$$

so higher scores correspond to lower entropy values. By definition, $0 \leq \hat{q}_i \leq 1$ so it can be plotted using a reliability diagram, however this calibrated score can no longer be interpreted as the probability of the input being of a particular class label. Naturally, the ECE at test time when using entropy as the calibration method is higher than the others as shown in fig. 6.1. Unlike previous methods, the confidence score when using the entropy method is not tied to the predicted class label.

Distance to Decision Boundary

The margin is the minimum distance to the decision boundary across all data points, and larger margins are desirable for the purpose of better generalization. We use the notion of distance to the decision boundary as a score indicative of the likelihood a particular input is correctly classified. Larger distances to the decision boundary can be associated with a greater chance of correct classification, whereas smaller distances can be associated with predictions that are likely incorrect. The distance to the decision boundary cannot be analytically computed for neural networks, unlike for more classical models like SVMs, so we use the method from Elsayed et al. [3] to approximate it.

For an input x, let h be the neural network where $h(x)$ returns a distribution $\begin{bmatrix} h(x)_1 & h(x)_2 & \dots & h(x)_k \end{bmatrix}$ over all k possible class labels. The predicted label is $\hat{y} = \arg\max \begin{bmatrix} h(x)_1 & h(x)_2 & \dots & h(x)_k \end{bmatrix}$. Let $h(x)_i$ and $h(x)_j$ correspond to the classes with the highest and second highest scores respectively. We will be taking distance of inputs to the decision boundary defined by these two classes.

The decision boundary can formally be defined as

$$D_{(i,j)} = \{x \mid h(x)_i = h(x)_j\}$$

Then, the distance for a input to the decision boundary under the l_p norm is

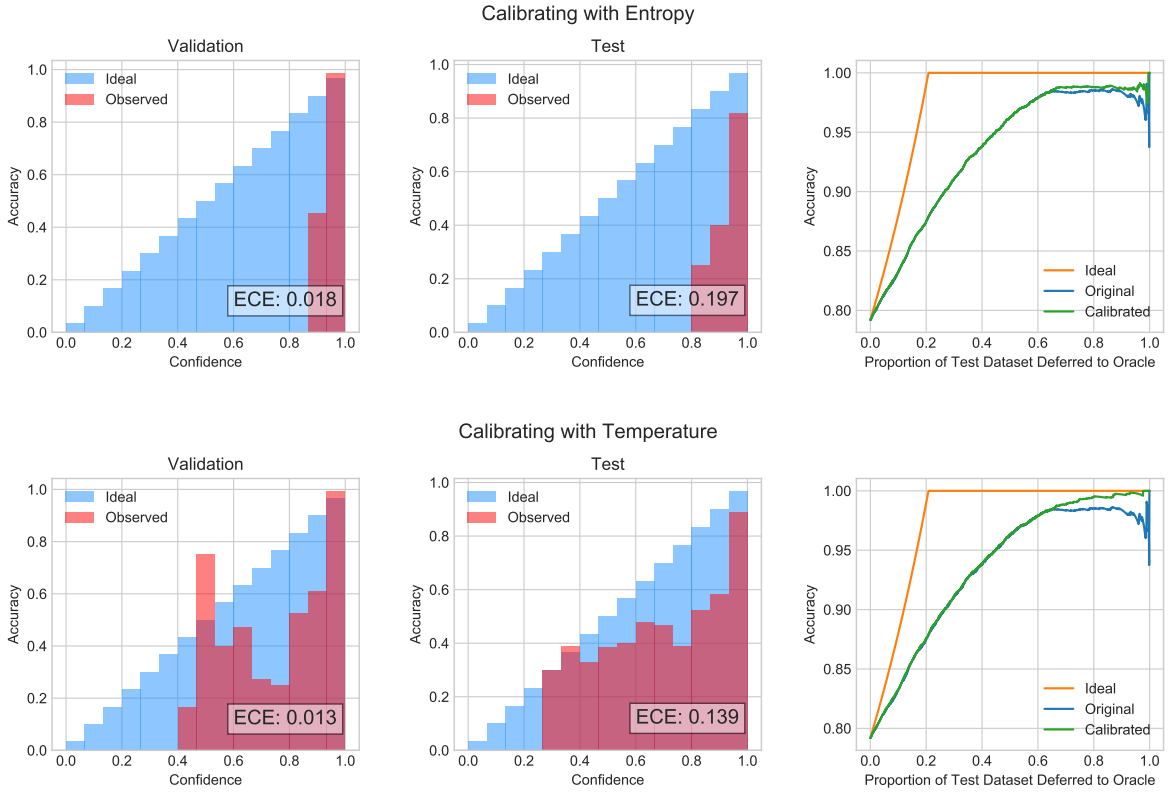
$$d_{(i,j)}(x) = \min \|\delta\|_p \quad \text{s.t.} \quad h(x+\delta)_i = h(x+\delta)_j$$

Because this equation is intractable, we take the first order Taylor approximation using the l_2 norm as described in Elsayed et al. [3]:

$$d_{(i,j)}(x) = \frac{h(x)_i - h(x)_j}{\|\nabla_x(h(x)_i) - \nabla_x(h(x)_j)\|_2}$$

We use this approximation of the distance directly as the score indicating likelihood of correct classification. Distances can be any positive value, so we analyze this method using the uncertainty class threshold diagram rather than the reliability diagram.

Discussion of Results



Recalibrating ResNet 44 on Cifar 10 to Probability				
	Calibrated Val ECE	Calibrated Val AAC	Calibrated Test ECE	Calibrated Test AAC
Histogram Binning	0.002	0.014	0.171	0.224
Entropy	0.018	0.001	0.197	0.064
Matrix Scaling	0.006	0.001	0.116	0.065
Temperature Scaling	0.013	0.001	0.139	0.061
Distance to Decision Boundary	—	0.001	—	0.062

Figure 6.1: Various calibrations of the confidence estimates are performed on ResNet-44/CIFAR-10 model. Without any calibration, the baseline test ECE is 0.168 and the baseline test AAC of the uncertainty class threshold diagram is 0.066. Entropy, Matrix Scaling, and Temperature Scaling seem to give more useful confidence values than the baseline of no calibration. Using entropy for the confidence estimate highlights the value of directly plotting the accuracy as a function of the proportion of data that is predicted on, because although the calibrated test ECE is higher than the baseline, the AAC is lower. Because the distance to decision boundary approximation doesn't result in a probability, the reliability diagrams and ECE cannot be calculated.

When comparing these methods, matrix scaling and temperature scaling perform best in terms of the ECE and the AAC for the uncertainty class threshold diagram respectively. In the case of both of these methods, we see that a good calibration of confidence estimates results in better separation of inputs that are likely incorrect. However, we also observe that the likely incorrect points can be flagged even without good confidence calibrations. The entropy method results in scores that cannot be interpreted as the estimated sample accuracy, and therefore has a ECE that is worse than the baseline. However, inputs that are likely incorrect seem to tend to have high entropy in the predicted probability distribution across classes, and can be flagged with this method. In addition, the distance to the decision boundary method results in scores that can be any positive number and don't have the interpretation of being the expected sample accuracy, so reliability diagrams and calibration errors cannot be computed.

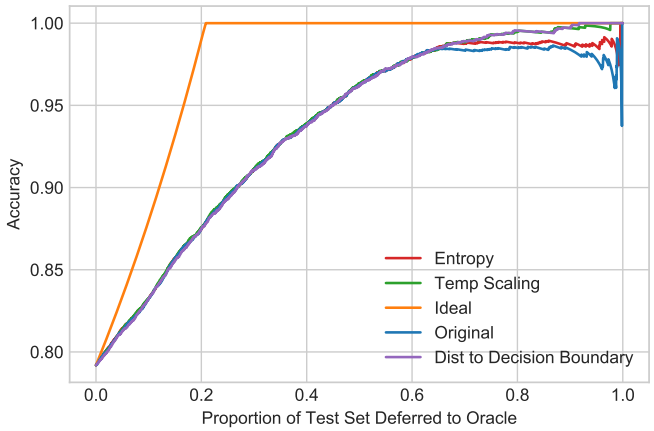


Figure 6.2: Plotting the proportion of the test set deferred to the expert vs the accuracy allows the comparison of methods that calibrate directly to expected sample accuracy and others that calibrate to general confidence scores. Reliability diagrams require that the calibrations are probabilities which can be interpreted as estimates of expected sample accuracy, so the ECE for methods like entropy and distance to decision boundary calibration are not meaningful.

Even though fig. 6.2 shows the distance to decision boundary method and temperature scaling generating similar curves on the uncertainty class threshold diagram, the temperature scaling method may still be a better choice because of its lower computational cost. It also has the benefit that the calibrated scores can be interpreted as expected sample accuracy, and can therefore be used in applications other than determining what subset of data to trust neural network predictions on.

Choosing Calibration Method - ResNet44/CIFAR10						
	Val Train ECE	Val Test ECE	Test ECE	Val Train AAC	Val Test AAC	Test AAC
Histogram Binning	0.004	0.006	0.163	0.013	0.022	0.235
Entropy	0.021	0.015	0.197	0.001	0.002	0.064
Matrix Scaling	0.007	**0.004**	**0.129**	0.001	0.002	0.071
Temperature Scaling	0.014	0.011	0.138	0.001	**0.001**	**0.062**
Distance to Decision Boundary	—	—	—	0.001	0.002	0.062

Figure 6.3: The correct calibration method to use for a given application can be determined by learning any parameters for calibration on a subset of the validation data and evaluating on the other subset. In this example the validation data was split in half for training and evaluating. Note the difference in some table values from fig. 6.1 are due to the calibration method being fit on only half of the validation data.

Although calibration methods like matrix scaling and temperature scaling tend to perform well, for a new problem, it is not immediately clear which calibration method would work best. Moreover, many of these methods have hyperparameters that affect the final performance. In fig. 6.3, we find that evaluating calibration methods on a subset of the validation set that is not used for training the learnable parameters for calibration gives a good indication of which method will perform well at test time.

Chapter 7

Conclusions and Future Work

Calibrated confidence estimates of predictions are critical to increasing our trust in the performance of neural networks. As interest grows in deploying neural networks in real work decision making systems, predictable behavior of the model will be a necessity. A confidence estimate that has the interpretation of the expected accuracy can be used as a value that is directly fed into the next stage of the decision making pipeline, whether that is another probabilistic model or a human expert acting on this information. For other applications, the confidence estimate may be used to decide whether to act on the model prediction, or defer this input to the expert.

Modern neural networks have been shown to be more overconfident in their predictions than predecessors even though their generalization accuracy is higher, partly due to the fact that they can overfit on the NLL loss without overfitting on the 0/1 error. We further investigate this idea by exploring the effect of various common regularization techniques on the calibration error. Tuning the label smoothing, weight decay, and dropout parameters and performing data augmentation seemed to have a positive impact on the ECE. However, the best value of some of the hyperparameters required for these regularization methods for the purpose of calibration were different than the best value to increase generalization accuracy, indicating tuning should be done using the ECE if that is a specific metric of interest.

The IDK class has been proposed as a method of allowing the model to choose not to predict on a particular input, but prior work on setting a threshold of when to choose this class hasn't explicitly accounted for the proportion of data that the model would defer to the oracle. We introduced the uncertainty class threshold diagram which visualizes across thresholds the proportion of data not predicted by the model against the accuracy that can be obtained by taking the best subset of data using a certain confidence scoring method. Calibration methods can be evaluated by taking the AAC of the threshold plot, with lower values indicating more useful confidence scores. This framework is naturally suited to applications with a high cost of misclassification in which there is a desire to gradually transition from a legacy

system to a machine learning model. Choosing a threshold based on this framework allows better control over the final accuracy of the system, and greater trust in its performance.

Although confidence estimates that are calibrated to equal the estimated sample accuracy are considered good calibration methods under this new framework, we find that confidence scoring methods that may not have good calibration errors can also perform well under this method of evaluation. Moreover the uncertainty class threshold diagrams don't require the confidence scoring method to even return a valid probability, because only the relative ordering of scores are used. We analyze entropy and introduce distance to decision boundary as two methods that don't directly estimate the expected sample accuracy. The method of scoring points by their estimated distance to the decision boundary showed performance that almost equaled that of temperature scaling. However, because of its comparative simplicity to implement and additional interpretation as an estimate of expected sample accuracy, temperature scaling is likely a better method of calibration to use in many settings. These experiments still highlight the value in using this alternate framework to analyze general confidence scores, rather than being limited to true confidence estimates when using reliability diagrams. If the goal of a confidence estimate is to develop this threshold, this framework can allow for more flexibility in the calibration methods and also open doors for confidence scoring that does more advanced transformations to return general confidence scores that may not even be probabilities.

There are a variety of methods that can be used to either calibrate the confidence estimates that the neural network predicts, or to modify the training procedure itself so the confidence estimates are better calibrated, so it is important that these methods can be compared using a framework that closely models the final use of the confidence estimates. For the purpose of determining a set of inputs that the model should not predict on to reach a desired accuracy, we proposed comparing the area above the curve of the uncertainty class threshold diagram on a portion of the validation set that was not used to fit any parameters for calibration.

Although this method can indicate which model would best indicate when it is likely to make an incorrect prediction, the accuracy on the validation set for a particular threshold on the confidence score is not necessarily equivalent to the accuracy one can expect at test time for that threshold. It is important for practitioners to be able to estimate what the model's accuracy will be before it is deployed, so further work could explore directly estimating what the test time accuracy if only a subset of the data is predicted on as determined by the validation set. However, there is still value to using this framework even if the test accuracy cannot be precisely estimated in advance. An analysis of the inputs that the model was not confident enough to predict on at validation time could give an indication of the types of inputs that would be deferred to the expert at test time.

This work assumes that the underlying distribution of data at train and test time are the same and determines thresholds accordingly, however there could be a covariate shift effect

or outliers that appear at test time that would change the relative threshold of when to not predict. There has been prior research on detecting these events and potentially correcting for them, but additional work can also be done on dynamically changing the threshold as the model predicts on individual inputs during deployment to maintain a certain accuracy.

Bibliography

[1] Rich Caruana et al. "Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission". In: *KDD '15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015).

[2] Morris H. DeGroot and Stephen E. Fienberg. "The Comparison and Evaluation of Forecasters". In: *Journal of the Royal Statistical Society. Series D (The Statistician)* 32.1/2 (1983), pp. 12–22. ISSN: 00390526, 14679884. URL: http://www.jstor.org/stable/2987588.

[3] Gamaleldin F. Elsayed et al. *Large Margin Deep Networks for Classification*. 2018. arXiv: 1803.05598 [stat.ML].

[4] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: (2015). eprint: arXiv:1506.02142.

[5] Bin-Bin Gao et al. "Deep Label Distribution Learning with Label Ambiguity". In: *CoRR* abs/1611.01731 (2016). arXiv: 1611.01731. URL: http://arxiv.org/abs/1611.01731.

[6] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *ICML*. 2017.

[7] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[8] Geoffrey E. Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *CoRR* abs/1207.0580 (2012). arXiv: 1207.0580. URL: http://arxiv.org/abs/1207.0580.

[9] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

[10] Xiaoqian Jiang et al. "Calibrating Predictive Model Estimates to Support Personalized Medicine". In: *Journal of the American Medical Informatics Association : JAMIA* 19 (Mar. 2012), pp. 263–74. DOI: 10.1136/amiajnl-2011-000291.

[11] Yiding Jiang et al. *Predicting the Generalization Gap in Deep Networks with Margin Distributions*. 2018. arXiv: 1810.00113 [stat.ML].

[12] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: (2016). eprint: arXiv:1612.01474.

[13] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. "When Does Label Smoothing Help?" In: *CoRR* abs/1906.02629 (2019). arXiv: 1906.02629. URL: http://arxiv.org/abs/1906.02629.

[14] Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. "Obtaining Well Calibrated Probabilities Using Bayesian Binning." In: *AAAI*. 2015, 2901–2907.

[15] Alexandru Niculescu-mizil and Rich Caruana. "Predicting Good Probabilities with Supervised Learning". In: *In Proc. Int. Conf. on Machine Learning (ICML*. 2005, pp. 625–632.

[16] John C. Platt. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[17] Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. *Learning for Single-Shot Confidence Calibration in Deep Neural Networks through Stochastic Inferences*. 2018. arXiv: 1809.10877 [cs.LG].

[18] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *CoRR* abs/1512.00567 (2015). arXiv: 1512.00567. URL: http://arxiv.org/abs/1512.00567.

[19] Sunil Thulasidasan et al. *On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks*. 2019. arXiv: 1905.11001 [stat.ML].

[20] T. P. Trappenberg and A. D. Back. "A classification scheme for applications with ambiguous data". In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. Vol. 6. 2000, 296–301 vol.6.

[21] Xin Wang et al. "IDK Cascades: Fast Deep Learning by Learning not to Overthink". In: *CoRR* abs/1706.00885 (2017). arXiv: 1706.00885. URL: http://arxiv.org/abs/1706.00885.

[22] Bianca Zadrozny and Charles Elkan. "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers". In: *ICML*. 2001.

[23] Bianca Zadrozny and Charles Elkan. *Transforming Classifier Scores into Accurate Multiclass Probability Estimates*. 2002.

[24] Chiyuan Zhang et al. "Understanding deep learning requires rethinking generalization". In: *CoRR* abs/1611.03530 (2016). arXiv: 1611.03530. URL: http://arxiv.org/abs/1611.03530.

[25] Hongyi Zhang et al. "mixup: Beyond Empirical Risk Minimization". In: *CoRR* abs/1710.09412 (2017). arXiv: 1710.09412. URL: http://arxiv.org/abs/1710.09412.