Delineating the Unknown:

Predictive Uncertainty Estimation a Posteriori

Iliana Maifeld-Carucci, M.S.
The George Washington University

Abstract

In recent years, machine learning has increased greatly in popularity and use; yet one big disadvantage of the field is the lack of predictive uncertainty estimates like is frequently found in most statistical methods. As machine learning algorithms become progressively more ubiquitous in application, it is important that users trust that their predictions reflect the true likelihood of that prediction or, said another way, that the prediction is well calibrated. This is advantageous to know, because if a machine learning prediction has a greater level of uncertainty, that prediction can be given greater attention, and mistakes can be avoided. Unfortunately, many of the most potent contemporary algorithms, Deep Neural Networks (DNNs), have not accomplished this aim. However, there have been a couple state-of-the-art methods developed recently to estimate predictive uncertainty in DNNs. Many researchers have focused on Bayesian Deep Learning (BDL) where a distribution is estimated over every parameter [3, 4, 5, 6, 7, 8]. Building on this, Lashminarayanan, et al. [10] found that using ensemble training with adversarial examples resulted in better predictive uncertainty estimates. Such estimates are defined as a measurement of how confident a model is in its output. Both of these methods, though, rely on white-box scenarios where the researchers are the ones estimating the model. In practice, researchers often encounter black-box situations where access to a model is not possible, only the model's output and ground truth. Can the predictive uncertainty of the unknown model's output still be estimated? This is the question this paper shall address. Using the absolute difference between the unknown model's output and true values as a target, another model will be trained with the same input, and will estimate the predictive uncertainty. These findings will be compared and contrasted with those from Lashminarayanan et al, and shows that the developed method yields comparable or better scores than those from our "unknown" base model, and the ensemble method.

Keywords: machine learning, neural networks, predictive uncertainty, calibration

Introduction

As the ability of computers to process large amounts of data has increased, machine learning has risen in usage and influence in order to gain insights from that data. In particular, neural networks (NNs) and, more recently, deep neural networks (DNNs) have achieved substantial advances in accuracy, and are being used in a variety of applications such as natural language processing and computer vision. Models such as these are being implemented in business and utilized in products like self-driving cars, facial recognition, augmented reality, healthcare diagnoses, fraud detection, and machine translation, among many others. As the applications of these machine learning techniques advance into our everyday lives, however, they make critical decisions that can seriously impact human lives. One can point to examples like Google's 2015 mistake in misclassifying two African-American people as gorillas [1], or the multiple fatalities from Tesla and Uber's self-driving cars that have occurred [2]. These impactful mistakes will, undoubtedly, increase as machine learning continues to infiltrate the technology people rely on. However, if there were a way to estimate when a machine learning system was unsure about its prediction, machine learning mistakes could be eliminated or better mitigated. That is why research into quantifying how confident an algorithm is in its prediction is critical as these algorithms become ubiquitous for use in daily life.

Unfortunately, unlike statistical methods, most neural networks do not incorporate uncertainty estimation, because the structure of the algorithms assumes that a single parameter generated the data distribution, and thus does not represent the distributions of parameters. Furthermore, their predictions are often overconfident which is equally, if not more, dangerous than when they are wrong. However, in recent years, a new wave of interest and research has emerged into quantifying various types of uncertainty in deep neural networks. Bayesian neural networks (BNNs) have received the most research attention since they are well-suited to uncertainty estimation inasmuch as they place a distribution over all parameters. Recently however, other methods have emerged, such as the use of ensemble techniques to predict uncertainty in DNNs. In situations where researchers have access to a model and its details, known as a white-box scenario, these methods provide substantial applicability, and estimate predictive uncertainty in a principled manner.

Nonetheless, in many practical circumstances, access to a model's specifics is not possible. This scenario is called a black-box case. Yet predictive uncertainty is still of concern. This paper will outline a method to assess uncertainty in such a black-box situation. Assuming access to an unknown model's inputs and outputs is available, this

research method will take the absolute difference between the unknown model's output predictions and its true target values, and then use those results as the target values for an attendant model. Then, using the same input features as the unknown model, the proxy model will try to predict this new target. This paper's method will combine these predictions with the unknown model's output predictions in order to create an adjusted set of predictions, which will then be scored by how closely the adjusted probabilities of a particular predicted class label reflect the underlying ground truth. The aim is for the ancillary model's output to adjust the original predictions, such that they move closer to their true class. This proposed technique will work because the auxiliary model is not predicting the same thing as the unknown model, but rather the deviation in predicted probability. Also, it is assumed that our objective function is not convex, which means that for some batches of observations, the unknown model will not output the optimal solution, and the ancillary model will capture the variance in the deviation of the unknown model's outputs.

Literature Review

The current research has developed a few main techniques for calculating predictive uncertainty in neural networks. The most common and widely researched of these so far is using Bayesian neural networks. These estimate a distribution over all parameters, thus allowing for estimates of the uncertainty surrounding predictions. The basis for much of this research was done in the late 1980's and early 1990's. More recently, researchers like Blundell et al. have renewed and built upon the prior techniques. In their paper [3], Blundell, et al., provided a backpropagation-compatible algorithm using Bayesian variational inference in approximating a posterior. By applying a reparameterization trick, they are able to get a variational estimation over the neural network's weights. In his dissertation, Gal [4] also uses Bayesian variational inference, but examines how Bayesian formalism can be combined with the strength and flexibility of deep learning to create more robust, trustworthy, and principled networks, which overcome some of the problems of traditional BNNs. Meanwhile, Hafner, et al. [5] improved on previous Bayesian analyses by changing the choice of prior from an uninformative standard normal prior to a noise contrastive prior, which adds noise to inputs and is designed to give better uncertainty estimates, particularly for data points outside of the training distribution. While these are all very promising methods that make BNNs more useful, BNNs are computationally intensive, and the training method must be adapted to handle the Bayesian estimation. Therefore, they have not had widespread adoption.

Extending Gal's thesis research, in 2016, Gal and Ghahramani [6] demonstrated the value of using dropout in DNNs as a Bayesian approximation in Deep Gaussian

processes. Dropout is the technique in neural networks wherein part of the nodes between layers are set to zero in order to prevent overfitting. Deep Gaussian processes are a statistical method to model distributions over functions, which allows for estimating uncertainty over function values, robustness to overfitting, and proper hyperparameter tuning. The state-of-the-art combination of these techniques that Gal and Ghahramani have developed eliminates the computational cost issue of Bayesian neural networks by directly working with a deep learning architecture, and shows significant improvement in predictive log likelihood over other prominent techniques. Proposing a variant to this original method, a year later Gal et al. [7] developed a dropout method that was able to be tuned faster and gave better calibrated uncertainty estimates. Drawing inspiration from using regularization techniques in neural networks as approximations of a Bayesian network, Teye et al. [8] show that batch normalization can also be cast as approximate Bayesian inference that offers competitive results. Batch normalization standardizes the distribution of each unit's input after each layer of a deep neural network, and improves generalization. These methods make significant strides over traditional BNNs by reducing the computational time and complexity; however, they are still only approximations of the Bayesian posterior that rely on some liberal assumptions.

Taking a different approach, Sensoy et al. [9] focus on uncertainty estimation in classification. By interpreting a softmax output as the parameter set of a categorical distribution and then replacing this parameter set with those of a Dirichlet density, they represent the predictions of a network as a distribution over potential softmax outputs instead of point estimates. They show that their method produces high quality uncertainty predictions particularly when applied to out-of-distribution data.

More recently, Lakshminarayanan, et al. [10] propose using an ensemble of DNNs to calculate predictive uncertainty. Ensemble training is a machine learning approach that provides a composite of several machine learning models to provide a more robust predictive capability with lower variance than a single model could usually accomplish alone. Using an ensemble of DNNs requires no adaptation of the training procedure as with BNNs, and outputs comparable or better uncertainty estimates than those generated from BNNs. However, their method models a Gaussian distribution at each feature location, which assumes symmetric and unimodal uncertainties.

When estimating uncertainty in deep neural networks, there are two main types. Aleatoric uncertainty deals with the noise inherent to the data while epistemic uncertainty quantifies the variability in a particular model. Aleatoric uncertainty can be broken down further into homoscedastic and heteroscedastic uncertainty. Homoscedastic uncertainty delineates instances when the uncertainty remains constant for different inputs. On the other hand, heteroscedastic uncertainty is used to describe cases where different inputs yield different levels of uncertainty. Most research does not

consider epistemic uncertainty, because it can often be dealt with by using more data and only focuses on aleatoric uncertainty. Bayesian neural networks have the ability to estimate both aleatoric and epistemic uncertainty; and in their paper [11], researchers Kendal and Gal presented a unified Bayesian deep learning framework that composed both forms of uncertainty. They characterize the properties of modeling each form of uncertainty and show the benefits and consequences of each. Additionally, they present a case for estimating heteroscedastic aleatoric uncertainty as a function of the data, which shall be further explored in this paper.

One measure of uncertainty that is closely related to aleatoric uncertainty is calibration which indicates how closely the probability associated with the predicted class label reflects its ground truth likelihood. Researchers Guo et al. [12] examine the calibration of multiple, well-known neural network architectures. They discovered that despite more impressive accuracy scores gained from modern architectures, these networks are actually much worse in terms of calibration than older networks. They come up with a variant of Platt scaling, called temperature scaling, and find that it is an effective post-processing method that can be used to obtain well-calibrated probabilities. While Guo et al. focus on multiclass classification, researchers Kuleshov et al. [13] extend their research to the regression problem, and found that it consistently outputs highly calibrated predictions.

Without a doubt, these methods increase the quality of predictive uncertainty in white-box situations; that is, in situations where researchers are the ones estimating the model. In real-world applications however, there are many scenarios where a researcher may not have access to a model, but where access to the inputs and outputs are available. In this paper, the focus will be on estimating aleatoric predictive uncertainty through calibration in a black-box situation where a model's details are not available. This research takes the absolute deviance between a base predictor's output predictions and the ground truth as the target variable for a secondary model. In essence, the ancillary model attempts to learn the predictive uncertainty of the first model. The last step combines the predicted probabilities from the base model with the predicted deviance of the auxiliary model into a new predicted probability that is better calibrated than the original.

Data and Analysis

This research will approach the estimation of predictive uncertainty in the binary classification problem through the use of the MNIST dataset. The MNIST dataset was assembled by the National Institute of Standards and Technology (NIST), and consists of grayscale, 28x28 pixel, normalized images of handwritten digits with 60,000 training examples and 10,000 test examples. This is a pre-cleaned dataset that is used

extensively in academic research, and thus minimal data munging is needed. While the original dataset has ten classes representing each number (zero to nine), because this paper solely examines the binary classification problem, it will only focus on ones and sevens. These two digits were chosen because they look very similar and are thus more difficult for a model to distinguish. Some variance in accuracy is desired due to the goal of creating an auxiliary model, which predicts the deviation between an unknown model's predicted and true values. Creating a subset based solely on these two numbers, the combined training and testing dataset is made up of 15,170 observations. The sevens' digit is reclassified to zero for purposes of classification, and all pixel values of the inputs are rescaled from the 0-255 range to the 0-1 range for faster training and easier convergence of the models.

Research Methodology

Because calibration is paramount to the estimation of uncertainty in this research, the choice of scoring rule (or outcome measure) for the models is important. By assigning a score to a predictive distribution, a scoring rule rewards better calibrated predictions more than worse ones. A proper scoring rule should evaluate the quality of the predictive distribution to be less than or equal to the true distribution. Such a scoring rule can then be incorporated into a neural network through minimizing it as the loss function. The negative log likelihood, which is a standard measure of probabilistic quality, also known as cross-entropy loss, is one proper scoring rule that shall be used. In the binary classification problem, a metric called the Brier score, which minimizes the squared error between the predicted probability of a label and its true label, is also a proper scoring rule and will be used as the main metric for this paper. The Brier score is a single number composition of three other measures calculated as reliability, minus uncertainty, plus resolution. Reliability is an overall measure of calibration that quantifies how close the predicted probabilities are to the true likelihood; uncertainty measures the inherent uncertainty in the predictions; and resolution assesses the degree to which predicted probabilities placed into subsets differ in true outcomes to the average true outcome. These last two measures, uncertainty and resolution, can also be aggregated into a measure known as 'refinement', which is associated with the area under the ROC curve and, when added to reliability, yields the Brier Score.

In this paper, three models are created: two dedicated to the proposed technique and one replicating the ensemble technique laid out in Lakshminarayanan et al. [10]. Model 1, or the base model, represents the unknown, black-box case. First, the dataset has been split such that 30% of the data is used for training, and 70% is for testing. While this split is highly unusual, it was purposefully chosen, since the absolute deviation between the base model's predictions and true values becomes the target for

model 3; and thus, the 70% testing set from model 1 will become the input for the third model, as can be seen in Figure 1. While the choice of model 1 architecture should not have an effect on the quality of the proposed technique, this paper uses a shallow convolutional neural network with 5x5 filters, two hidden layers, and a sigmoid activation function. Regularization techniques such as dropout or batch normalization were not used, in alignment with the research findings of Guo et al. [12], who demonstrate that these additions can decrease calibration.

Model 2 is the ensemble model that replicates the approach of Lakshminarayanan et al. [10]. Here, the same sample of data using the 30/70 train test split from the base model is used with the identical architecture from model 1; but instead of only training one model, a set of five is trained. Although the architecture of each network is identical, because of different initializations and estimations of the objective function, the output predictions are different. The predictions of each model are subsequently stored; and after all models have returned outputs, the set of predictions for each input is averaged and the Brier score is assessed.

As previously described, model 3 represents our auxiliary deviation model, which takes in the 70% of testing data, and attempts to predict the absolute difference between the base model's predicted probabilities and the base model's true targets. Since this deviation is continuous, model 3 becomes a regression problem. It is also important to note that because this absolute difference lies in the range between zero and one, and model 3 is predicting a regression, a logit transform must be applied to the target values so that their range is all real numbers. Model 3 does not use a train test split, since the goal is not to use the model for accuracy, but to adjust the predictions from model 1. The same architecture as in model 1 is again used; however, in this case, more neurons and a linear activation function has been added. The loss function also changes from negative log likelihood to root-mean-squared error (RMSE), a scoring rule better suited to regression problems. Once model 3 generates predictions, a logistic transformation is used to return the predictions to the zero-to-one range in order to be combined with model 1's output. However, for the Brier Score to be applied, these combined and adjusted predictions must also fall in the range of zero to one. Three cases emerge, which ultimately shape the adjustment function used to alter the predictions from model 1. For predictions from model 1 (m1) and model 3 (m3) respectively, the adjustment function is

$$f(m1,\ m3)\ =\ 1/2((1-m3)+m1)\ for\ m3<.5\ and\ m1>.5;\ 1/2(m3+m1)\ for\ m3,\ m1<.5;\ .5\ for\ m3>.5$$

. The Brier score is then applied to the adjusted predictions.

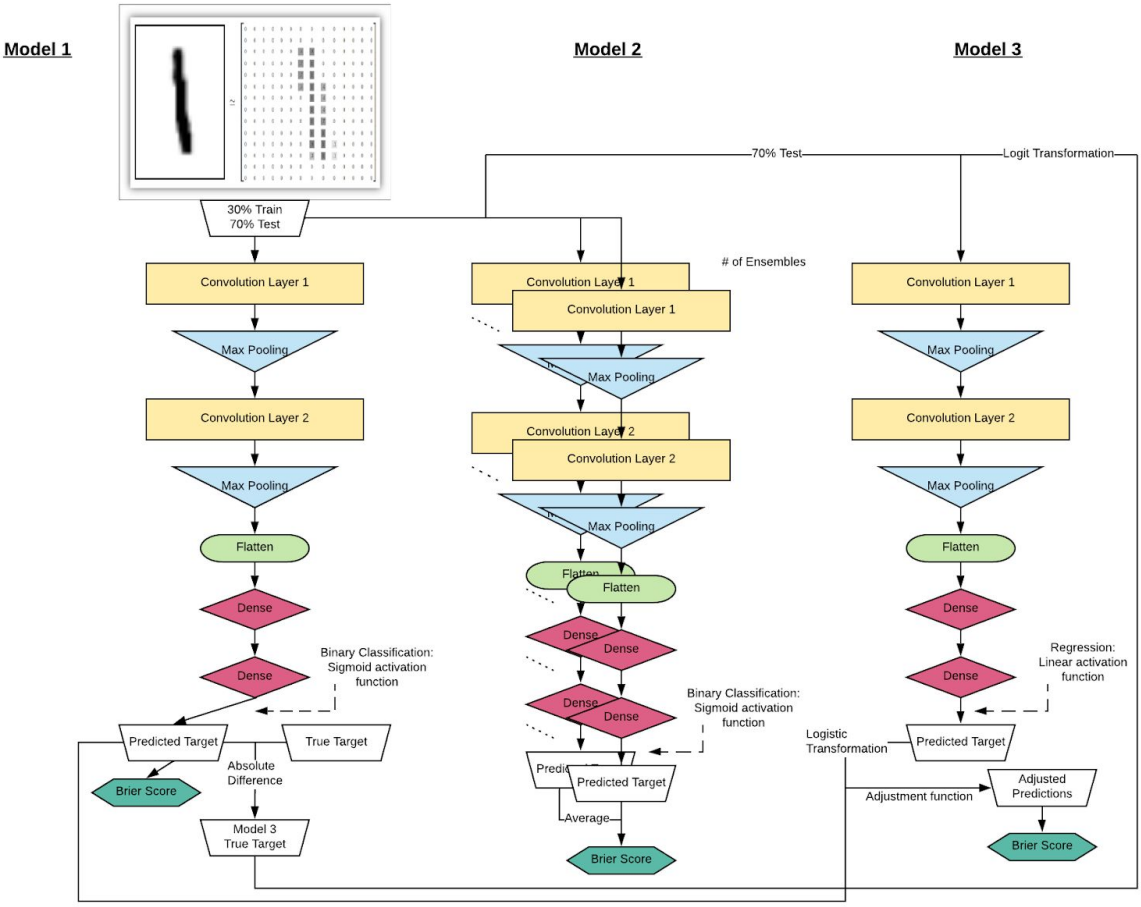

Figure 1

Key Findings

Running model 1, it can be seen in Figure 2 that the loss and Brier score quickly move toward zero while the accuracy is very high, indicating that this model has a low level of uncertainty with a Brier score of 0.00694, a high classification accuracy of .9910, and a low loss of .0310. This is confirmed in the calibration plot in Figure 3, which shows that model 1 is fairly well-calibrated except at areas around .3 and .6 probability, which are overconfident. The density plot in Figure 4 shows the output prediction distribution by class, and indicates the model is doing a good job of outputting values close to the class label since there are strong peaks near zero and one.

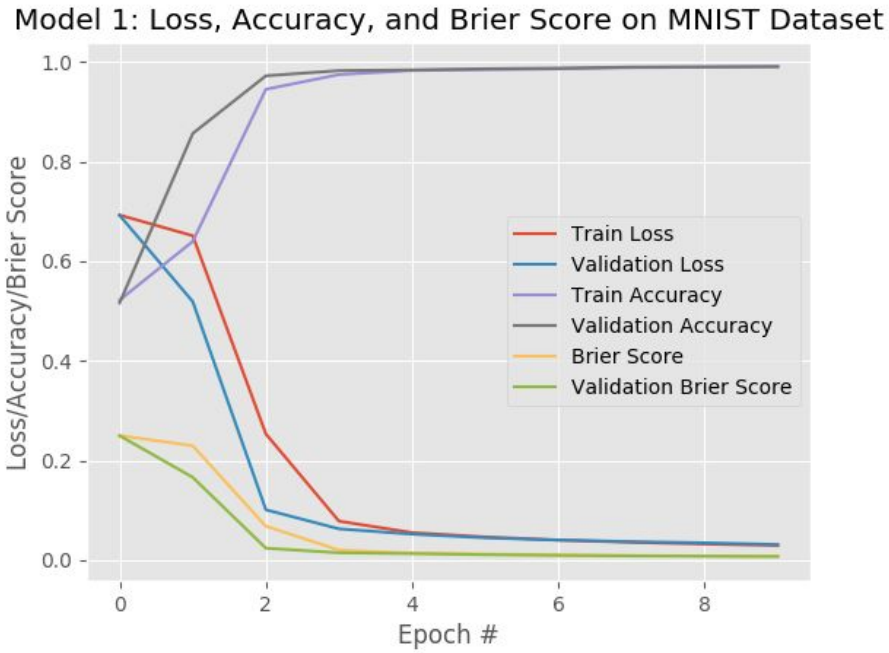

Figure 2

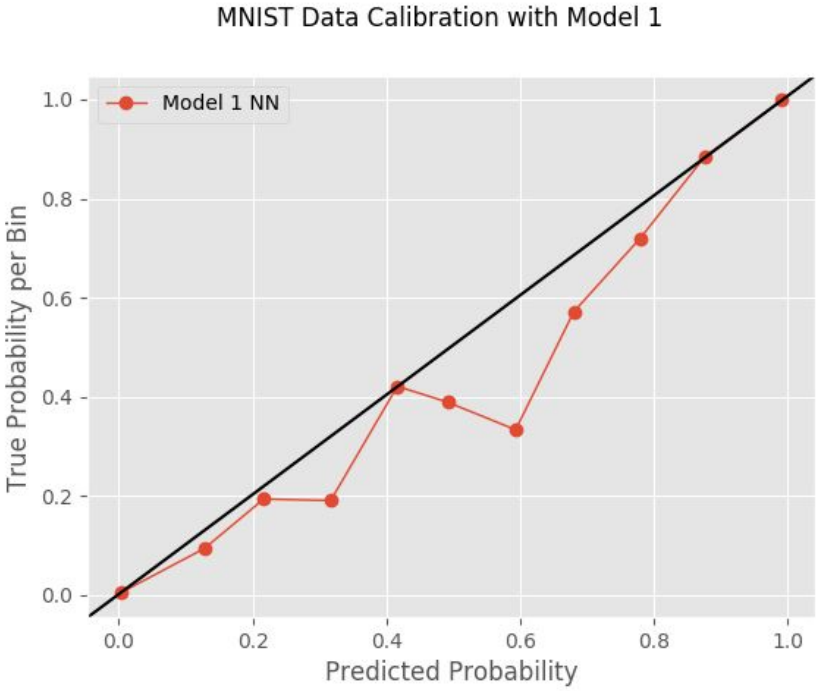

Figure 3

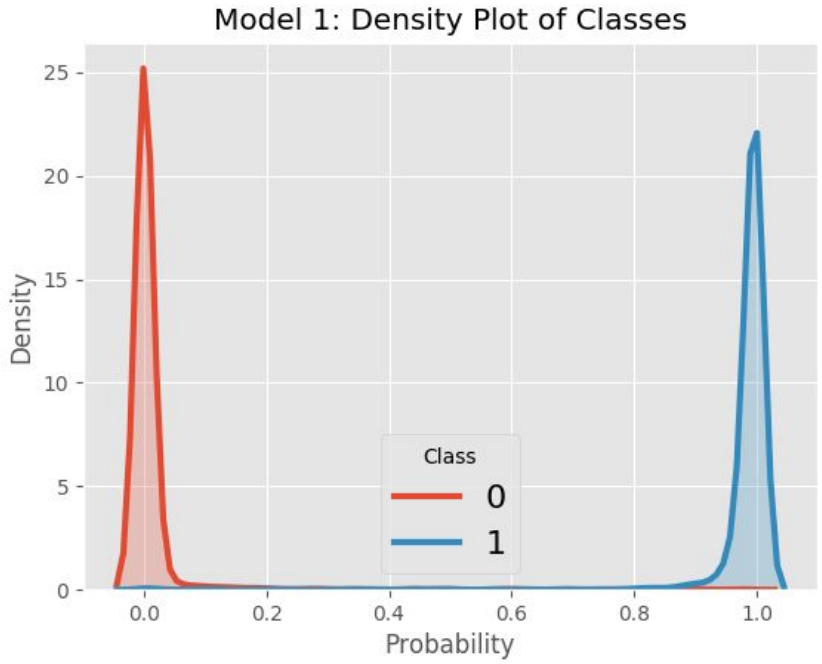

Figure 4

A rather different picture emerges after completing the computation of the ensemble of networks in the architecture of model 2. The calibration plot in Figure 5 indicates that when the predicted probabilities of a set of models are averaged, the calibration goes down with all predicted probabilities under .5 being overconfident, while predictions over .5 are underconfident; this is corroborated by an increased Brier score of 0.05366. The density plot in Figure 6 also reflects the higher predictive uncertainty as the peak of the predictions for each class has moved away from their true class towards .2 and .8 for class zero and one, respectively, and widened in distribution.

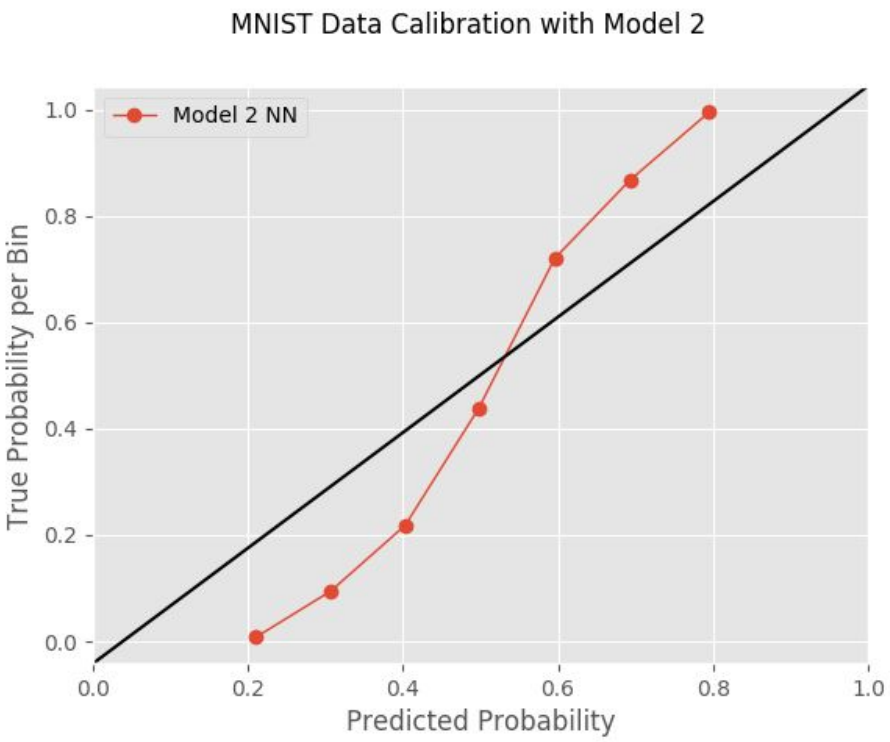

Figure 5

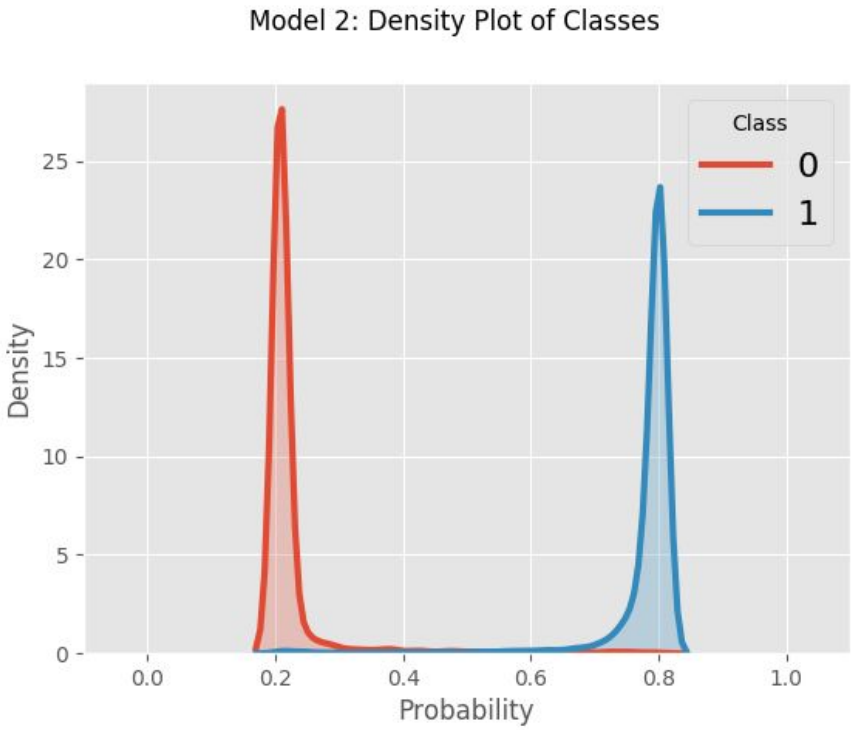

Figure 6

Finally, after processing model 3 and combining its output predictions with those of model 1, a comparable picture of calibration to that of model 1 is discovered with a

Brier score of 0.00685. This should not be particularly surprising, since the adjusted probabilities include those of model 1; but the way that the calibration changes is interesting, as shown in Figure 7. As can be seen, all predicted probabilities are underconfident, except those above .8 are very overconfident. Also, it is evident that there are no adjusted probabilities in the .4 to .6 range, which means the adjustment function has pulled the probabilities towards the more extreme values. This trend can be more clearly seen in Figure 8, which plots the predicted probabilities of model 1 versus the adjusted probabilities of model 3, color coded by their true class. The density plot in Figure 9 also reflects a very similar distribution of each class to that of model 1. With predictive uncertainty scores that are so similar to those of model 1, and superior to those of the ensemble technique in model 2, it can be said that the method proposed in this paper can be successfully utilized in situations where predictive uncertainty is desired, but a model's specifics are unknown.

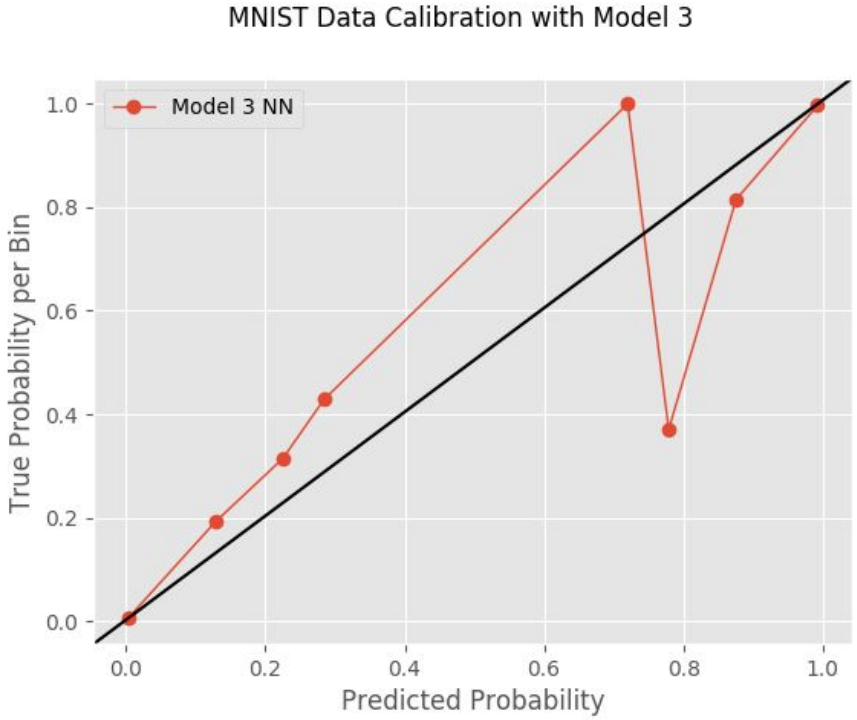

Figure 7

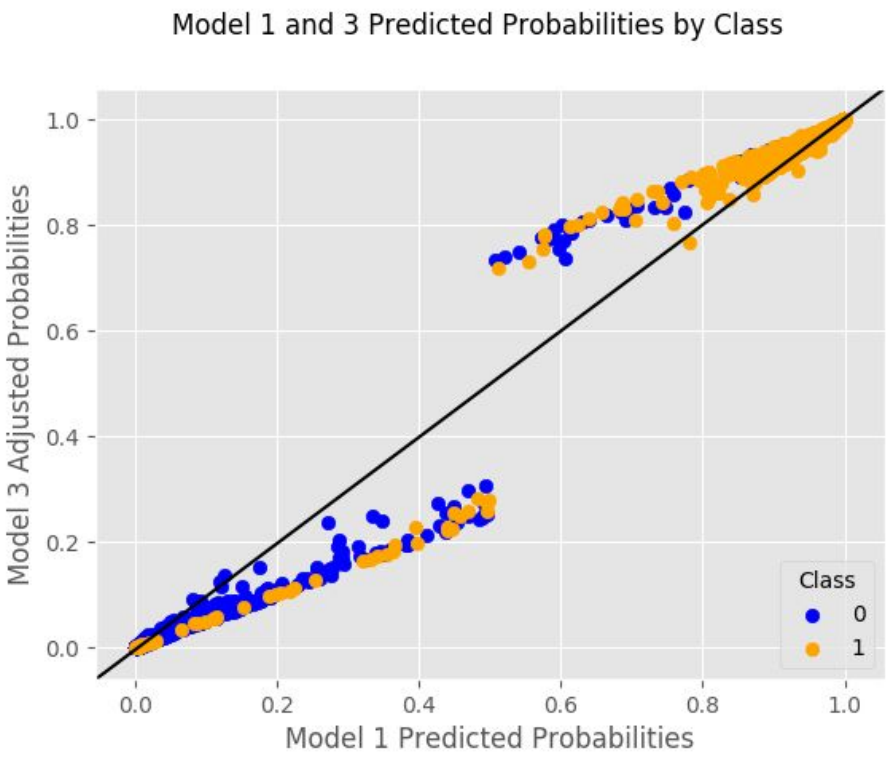

Figure 8

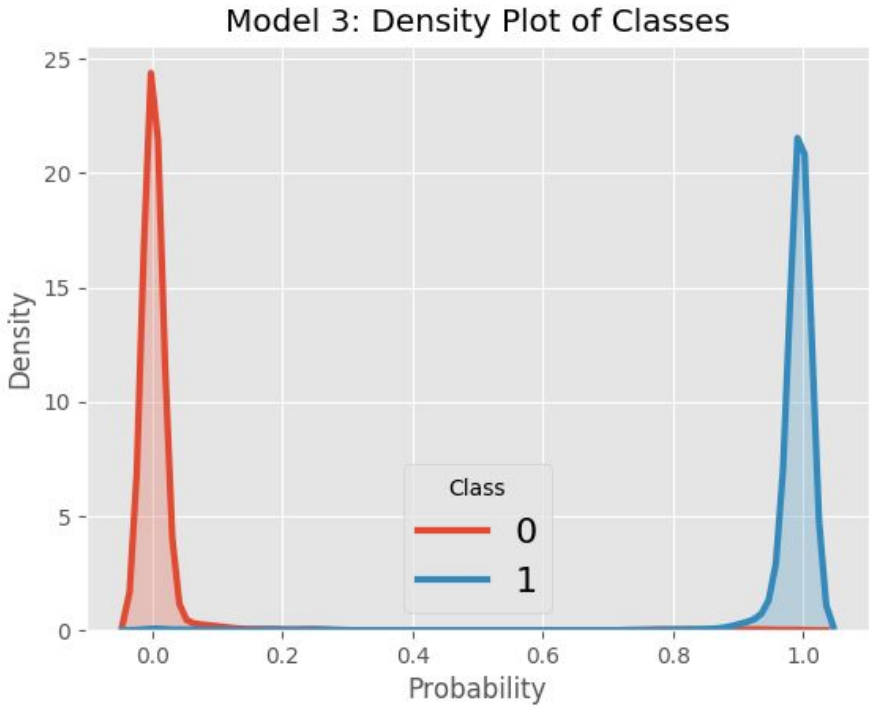

Figure 9

Future Research

There are multiple aspects of future research that can be elaborated upon based on the technique outlined in this paper. First and foremost, a more robust adjustment function could be constructed that moves predictions towards better calibration and lower predictive uncertainty than the one created and used in this work. Furthermore, this paper only discusses the binary classification problem on one dataset; therefore, research could be done to expand the rigor of the technique as well as its application to other types of problems such as multiclass classification or regression. Exploring other types of uncertainty such as testing this method against out-of-distribution data would additionally add context. It would also be interesting to see if using this technique in an ensemble situation such as that of model 2 would increase the quality of the adjusted probabilities, making them more finely calibrated.

Conclusion

As a discipline, and in practical application, machine learning and, more specifically, DNNs, have shown impressive predictive accuracy on a broad range of problems, and hold substantial promise for future application. However, not being able to quantify how much and when they are uncertain in their prediction in DNNs is a severe drawback to more widespread adoption. The established research approaching this problem has been encouraging, but it only applies to white-box situations. In practice, such situations are not always applicable. This research has tackled the problem of predictive uncertainty in a black-box scenario when a model is unknown. It has demonstrated that by using an auxiliary model to predict the deviation of the black-box model's outputs from its true values, and combining these predictions with the unknown model's outputs, the calibration of the ancillary model was comparable to that of the black-box model. Furthermore, it performed better than prior research using an ensemble method to predict uncertainty. Refining and extending this technique has the potential to provide good predictive uncertainty estimates in situations when quantifying the level of a model's uncertainty is desired, but details of the model are not known.

Biography

Iliana Maifeld-Carucci is an accomplished and ambitious master's student of data science, specializing in machine learning at George Washington University. As a data sleuth, who is always up for a challenge, the key to her success is her passion for using skills like data mining, statistics, machine learning, R, and Python to power change across industries. She has developed a reputation for thoroughly examining the assumptions behind every research project, and being a rising tide that elevates the knowledge of fellow peers. As a numbers geek and Argentine tango dancer, she applies the cultural intelligence gained from travel and living abroad, as well as her experience analyzing data from different angles, to ensure high-quality, innovative, and ethical data science solutions.

Bibliography

[1] Grush, Loren. “Google Engineer Apologizes after Photos App Tags Two Black People as Gorillas.” The Verge, The Verge, 1 July 2015, www.theverge.com/2015/7/1/8880363/google-apologizes-photos-app-tags-two-black-people-gorillas.

[2] Yadron, Danny, and Dan Tynan. “Tesla Driver Dies in First Fatal Crash While Using Autopilot Mode.” *The Guardian*, Guardian News and Media, 30 June 2016, www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk.

[3] Blundell, Charles, et al. "Weight uncertainty in neural networks." *arXiv preprint arXiv:1505.05424* (2015).

[4] Gal, Yarin. *Uncertainty in deep learning*. Diss. PhD thesis, University of Cambridge, 2016.

[5] Hafner, Danijar, et al. "Reliable uncertainty estimates in deep neural networks using noise contrastive priors." *arXiv preprint arXiv:1807.09289* (2018).

[6] Gal, Yarin, and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning." *International Conference on Machine Learning*. 2016.

[7] Gal, Yarin, Jiri Hron, and Alex Kendall. "Concrete dropout." *Advances in Neural Information Processing Systems*. 2017.

[8] Teye, Mattias, Hossein Azizpour, and Kevin Smith. "Bayesian uncertainty estimation for batch normalized deep networks." *arXiv preprint arXiv:1802.06455* (2018).

[9] Sensoy, Murat, Lance Kaplan, and Melih Kandemir. "Evidential deep learning to quantify classification uncertainty." *Advances in Neural Information Processing Systems*. 2018.

[10] Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." *Advances in Neural Information Processing Systems*. 2017.

[11] Kendall, Alex, and Yarin Gal. "What uncertainties do we need in Bayesian deep learning for computer vision?." *Advances in neural information processing systems*. 2017.

[12] Guo, Chuan, et al. "On calibration of modern neural networks." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

[13] Kuleshov, Volodymyr, Nathan Fenner, and Stefano Ermon. "Accurate uncertainties for deep learning using calibrated regression." *arXiv preprint arXiv:1807.00263* (2018).