LEARNING TO ESTIMATE EPISTEMIC UNCERTAINTY IN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Epistemic uncertainty quantification provides useful insight into both deep and shallow neural networks' understanding of the relationship between their training distributions and unseen instances and can serve as an estimate of classification confidence. Bayesian-based approaches have been shown to quantify this relationship better than softmax probabilities. Unfortunately, however, those approaches to uncertainty quantification require multiple Monte-Carlo samples of a neural network, augmenting the neural network to learn distributions for its weights, or utilizing an ensemble of neural networks. Such extra calculations are problematic in time- and/or resource-limited scenarios such as trauma triage and edge computing. In this work, we propose a technique that allows epistemic uncertainty to be estimated using learned regression algorithms. We find that this technique, once trained, allows epistemic uncertainty to be effectively and efficiently predicted.

1 INTRODUCTION

Neural networks have produced state-of-the-art results in a variety of domains; however, their use in safety-critical domains are limited due to their black-box nature and complex architectures (Begoli et al., 2019). This has paved the way for research in uncertainty quantification and explainable artificial intelligence. Both of which strive to improve trust between human practitioners and their artificial intelligence counterparts.

Epistemic uncertainty provides a numerical measure of how similar an input instance is compared to data on which the neural network has been trained and can be used to effectively indicate to human practitioners which instances should be referred to domain experts (Brown & Talbert, 2019; Leibig et al., 2017). This comes at the cost, however, of augmenting neural network architectures and requiring multiple, expensive inferences of a large neural network. This underlies the motivation for more efficient epistemic uncertainty estimates that can ascertained in production.

1.1 MOTIVATION

There are two primary scenarios that motivate this work: First, we need to consider time-critical domains, such as trauma triage or intensive care. The second scenario is that of edge-computing using field-programmable gate array (FPGA) devices.

Trauma triage and intensive care are subfields within medicine that can benefit from advances in deep learning. Trauma and emergency department triage (Sasser et al., 2012; Cole et al., 2016) categorizes patients based on the severity of their injuries or illness to allow medical professionals to attend to more critical patients first. Machine learning has not been extensively applied or studied in the trauma and emergency departments (Liu & Salinas, 2017). One reason for this is the black-box nature of neural networks. Uncertainty quantification is one technology that can aid in trauma and emergency department triage. Because of their time-critical nature, however, it is important to eliminate avoidable delays in decision support algorithms. Currently, state-of-the-art approaches to epistemic uncertainty quantification require, at the least, multiple inferences of a deep neural network which can be both expensive to compute and difficult to parallelize.

The second motivating example is the use hardware encoding of deep neural networks on FPGA devices. Encoding a deep neural network on an FPGA device allows for more rapid inference

and easier application of deep neural networks in edge computing cases, such as autonomous vehicles (Hadidi et al., 2019). A drawback to hardware encoding, however, is that dropout, a necessary operation for calculating epistemic uncertainty, is difficult to implement due to its stochasticity (Sawaguchi & Nishi, 2018; Myojin et al., 2020; Li et al., 2019). Using an auxiliary regressor to estimate epistemic uncertainty could allow epistemic uncertainty values to be computed on the FPGA device itself without the need to implement dropout or other Bayesian neural network architectures.

1.2 Contributions

We present the following contributions with this work: First, we propose a technique that allows epistemic uncertainty to be estimated through machine learning regression. Then, we assess this technique using multiple datasets and show that it yields high goodness-of-fit and produces uncertainty estimations comparable to common techniques such as dropout. This technique allows uncertainty estimates to be computed using few machine learning inferences.

The structure of this work is as follows: In the background, we describe the importance of epistemic uncertainty along with drawbacks to current approaches. We then present our technique and experimental methodology, followed by our results and the corresponding discussion before concluding this work.

2 Background

2.1 Uncertainty Quantification in Neural Networks

There are two broad classes of uncertainty in neural networks: *epistemic uncertainty* and *aleatoric uncertainty* (Kendall & Gal, 2017). Aleatoric uncertainty in refers to the extent to which noise in the data reduces the effectiveness of a learned classification or regression model (Kendall & Gal, 2017). This type of uncertainty can only be mitigated by reducing the noise in the data and is generally considered irreducible at the algorithmic level. This work, however, is concerned with epistemic uncertainty in neural networks.

2.1.1 Why Epistemic Uncertainty is Fundamental

Epistemic uncertainty gives an estimate of where, within the data distribution, a data point of interest lies (Gal & Ghahramani, 2016). Higher epistemic uncertainty implies greater distance between the data point of interest and the training data distribution.

Neural networks, as all machine learning algorithms, require a training distribution from which to infer a generalizable classification or regression function (Mitchell, 1997). As data to predict strays from this training distribution, predictions become more unreliable since the machine learning model is being asked to extrapolate into a previously unseen region of the data.

For many classification tasks, neural networks employ the softmax function to convert neuron outputs into prediction probabilities. Gal & Ghahramani (2016) notes, however, a fundamental flaw in that function. For data that lie outside the training data distribution, softmax tends to yield an overly confident probability regardless of the reliability of the classification. This is apparent in many foundational papers discussing adversarial examples and adversarial attacks (Goodfellow et al., 2014).

2.1.2 Measuring Epistemic Uncertainty in Neural Networks

True epistemic uncertainty can be calculated through a *Gaussian process*. By definition, a Gaussian process is a collection of random variables that have normal distribution (Rasmussen, 2003). In machine learning, a Gaussian process learns the appropriate mean and covariance functions to output a normal distribution of possible output values for given inputs. Epistemic uncertainty in a Gaussian process, then, is the variance of the output distribution (Rasmussen, 2003).

Work by Williams (1997) and Lee et al. (2017) have mathematically related a Gaussian process to a neural network. A neural network with a single layer (i.e., a shallow neural network) has been proven to be equivalent to a Gaussian process given the single hidden layer has an infinite number of neurons (Williams, 1997). Lee et al. extends this result to deep neural networks by showing multiple

hidden layers with infinite neurons converge to a Gaussian process (Lee et al., 2017). Since this is computationally infeasible, approximations are required.

The most accessible technique to calculate epistemic uncertainty in a deep neural network is to use dropout to induce an ensemble (Gal & Ghahramani, 2016). Dropout is a regularization technique that randomly removes neurons from layers within the neural network (Srivastava et al., 2014). When activated during training, dropout helps to prevent overfitting. Gal & Ghahramani (2016) showed however, when active during testing or production, dropout can be used to calculate epistemic uncertainty by performing repeated Monte-Carlo samples and calculating the variance of the predicted probabilities.

Bayes by Back-propagation (Blundell et al., 2015) takes an alternative approach to uncertainty quantification. This approach views network weights not as singular learned points, but as learned distributions. The weights of the algorithm are drawn from a Gaussian distribution and augmented using parameters μ and ρ. Bayes by Back-propagation learns the μ and ρ parameters rather than the individual weights of the network using an altered version of traditional back-propagation.

Lakshminarayanan et al. (2017) proposed another method of measuring uncertainty using deep ensembles. The deep neural network architectures within the ensemble are randomly generated to help reduce the correlation between models, and the final prediction and uncertainty value are ascertained through calculating the average (prediction) and the variance (uncertainty) of the outputs of the model. The ensemble is trained using data augmented by adversarial examples based on the training data. Lakshminarayanan et al. (2017) argue that the use of adversarial examples make both the predictor and uncertainty values more robust.

2.2 Regression Algorithms

Regression is a supervised learning task that predicts a numerical value, such as housing prices, rather than a classification probability (Mitchell, 1997). Many machine learning techniques such as decision trees, tree ensembles, and neural networks can be used for regression tasks.

2.2.1 Tree-based Regression

Decision trees are popular machine learning algorithms for both classification and regression tasks due to their inherent explainability (Breiman et al., 2017). Decision trees are designed to reduce entropy among the samples to create split points. In regression, the split points of each node are determined based on the minimization of standard deviation or measure of regression error. Predictions are computed by averaging the samples remaining at each leaf.

The primary advantage of a decision tree over other techniques such as neural networks, support vector machines, or boosting trees is their natural interpretability (Cooper et al., 1997). Decision trees produce a logical, tree-based structure that can be followed by human practitioners and domain experts.

We also consider gradient boosting trees as a regression technique for estimating epistemic uncertainty. Gradient boosting trees sequentially build decision trees, with subsequent trees focused on correcting the mistakes of prior trees (Friedman, 2002; Chen & Guestrin, 2016). Gradient boosting trees are considered one of the most successful and widely used current machine learning algorithms (Bennett et al., 2007; Xu et al., 2021).

2.2.2 Neural Network-Based Regression

Neural networks can also be applied to regression tasks. A neural network is a structure inspired by neurobiology and consist of interconnected "neurons" that propagate signals (Specht, 1991). The values of a neuron are the result of a linear combination the values of neurons in the previous layers and the weights that connect the previous layer's neurons to the current neuron. The value of this operation is then provided to an activation function that determines if the neuron "fires" or not (Rumelhart et al., 1985). Common activation functions include the sigmoid function (Rumelhart et al., 1985; Specht, 1991), the rectified linear unit (Nair & Hinton, 2010; Maas et al., 2013), and the hyperbolic tangent function (Glorot & Bengio, 2010). For regression, the final layer that produces the regression values does not include an activation function.

3 PREDICTING EPISTEMIC UNCERTAINTY

In this section, we discuss our proposed methodology to produce rapid, inferred uncertainty estimates based on regression algorithms. We also present benefits to using regression to model neural network epistemic uncertainty.

3.1 REGRESSION TRAINING METHODOLOGY

Let f be a machine learning regression algorithm, either multi-class regressor or multiple single-class regressors. To infer epistemic uncertainty using f, we use the original input features into the classification neural network and the output probability vector of predictions from the classification neural network. Thus, for an m-class classification problem, if $\vec{x} \in \mathbb{R}^n$ is the input vector of n features, $\vec{p} \in [0,1]^m$ the vector of output probabilities from classification network, then the vector of uncertainty estimates $\vec{u} \in \mathbb{R}^m$ is calculated as $\vec{u} = f(\vec{x}, \vec{p})$. We include $\vec{p}$ since uncertainty estimates should be conditioned on both the input data and the classification probabilities produced by the classification neural network.

Successfully training the uncertainty regression model requires augmenting the classification neural network's training data. When training the classification neural network, we use a *classification training set* to determine model weights through backpropagation, a *classification validation set* for model selection, and a *test set*. The regression-based uncertainty estimator is trained on the *regression training set* and evaluated using the *test set*. The regression training set combines both the classification training set and the classification validation set along with their classification probability vectors. These are then labeled with their dropout uncertainty. This enables the regression model to learn from examples outside of the classification training set's data distribution. The regression test set uses the same data as the classification test set.

3.2 NEED FOR APPROXIMATION

In addition to considerations discussed in the Introduction and Background, another consideration as to why inferred epistemic uncertainty values would be beneficial compared to current epistemic uncertainty quantification techniques is that of explainability and interpretability. Little to no work exists that algorithmically explains what *causes* uncertainty in deep learning. Although epistemic uncertainty can be reduced by training the neural network on additional data, the type of data needed to do so (e.g., under-represented features or combinations of features) is unknown. Decision tree regressors provide inherent interpretability within its structure, and new explainable artificial intelligence (XAI) algorithms are a very pertinent and growing subfield of Artificial Intelligence.

4 EXPERIMENTAL METHODOLOGY

In this section, we discuss experimental methodology starting with datasets and their respective classification models. We then discuss the implementation details, performance metrics, and experiments to evaluate our proposed approximation technique.

4.1 DATASETS AND CLASSIFICATION MODELS

We use three datasets in our experiments. Two datasets (Adult Income and MNIST) are classic benchmarking datasets for machine learning algorithms, and the third dataset is real-world critical-care dataset. We detail each dataset below as well as the classification models trained for their respective classification tasks.

4.1.1 ADULT INCOME

The first dataset is a modified version of the Adult Income dataset from the University of California at Irvine (UCI) Machine Learning Repository (Dua & Graff, 2017). This dataset contains a binary classification task to predict whether a person has an income greater than or less than $50,000. Features in this dataset include population demographics including age, gender, race, and marital status as well as occupation information including level of education, occupation type, work class

Table 1: Training information for the classification neural networks by dataset

Dataset	Hidden Layer Architecture	No. Training Epochs
Adult Income	20, 8	100
MNIST	128	50
Trauma Triage Registry	75, 75	100

(e.g., government, private, self-employed), and hours per week. Categorical features are represented using one-hot encoding. Continuous features are normalized using z-values.

4.1.2 MNIST

The second dataset is the MNIST dataset (Deng, 2012), which contains images of size 28×28 that depict handwritten numerical digits between 0 and 9. The multi-class classification task is to determine which digit is depicted per image. The features used by the classification neural network are the pixels of the 28×28 image that are converted into a one-dimensional vector of length 784. The pixel values are normalized by dividing each by 255 (the maximum pixel value).

4.1.3 Trauma Triage Registry

The final dataset is a subset of a trauma registry from a Level 1 Trauma Center that uses 32 features to determine if a patient has an Injury Severity Score (ISS) of at least 15. An ISS $\geq$ 15 indicates the patient is severely injured and should be triaged as such (Sasser et al., 2012). Features include physical parameters (e.g., systolic/diastolic blood pressures, heart rate, Glasgow Coma Scale score), anatomical criteria, mechanism of injury, age, and multiple computed injury scores (e.g., Revised Trauma Score and the Air Medical Prehospital Transport Score). Categorical features are represented using one-hot encoding. Continuous features are normalized using z-values.

4.1.4 Classification Models

For each classification task, we train a fully-connected neural network to learn a predictive model. Each dataset is split into 10 cross-validation folds with 10% of the training data in each fold used as a validation set during training of the neural network. The number and depth of the hidden layers as well as the number of training epochs are in Table 1. Hidden layers use the Rectified Linear Unit (ReLU) as their activation functions, and each output layer uses the softmax function as its activation function. Dropout is applied before each hidden layer to calculate epistemic uncertainty (Gal & Ghahramani, 2016). The ADAM optimization algorithm (Kingma & Ba, 2014) is used for the MNIST dataset while the RMSprop optimization function (Hinton et al., 2012) is used for the Adult Income and Trauma Triage Registry datasets. We use the Keras API (Chollet et al., 2015) with Tensorflow (Abadi et al., 2015) to implement all neural networks in this work.

4.2 Regression Algorithms and Training Details

We used three machine learning algorithms to model epistemic uncertainty. First, we use the CART decision tree algorithm as implemented in Scikit-Learn (Pedregosa et al., 2011) with squared-error as the split criterion.

For boosting trees, we use the XGBoost Python (Chen & Guestrin, 2016), with standard parameters for an XGBoost regression task (*maximum tree depth*: 16; *learning rate*: 0.3; *objective function*: linear regression; *loss function*: mean absolute percent error; *L1 regularization term* α: 10).

We use Keras and Tensorflow (Chollet et al., 2015; Abadi et al., 2015) to implement the regression neural networks. For each dataset, we consider a network with hidden layers of width 512, 255, 128, 64 neurons. We used mean squared error as the regression loss function and the ADAM algorithm as the optimization function (Kingma & Ba, 2014). Each network is trained for 50 epochs.

For decision tree- and neural network-based regressors, we consider both learning one uncertainty prediction model per possible class (referred to as *multi-single* model) and a single model that produces uncertainty information for all classes simultaneously (referred to as *multi-output* model).

4.3 Performance Metrics and Experiment Design

We split each dataset into 10-folds for cross-validation. The training set for each fold is further divided into training and validation.

We report several evaluations metrics. First, we report baseline accuracy for each classification neural network. Then, for regression, we report the R^2 correlation coefficient between dropout uncertainty and the predicted uncertainty from our technique on the testing data of each fold. This will inform us of the goodness-of-fit in our regression models. We also report mean absolute percentage error of the regression model on the unseen data.

To evaluate the uncertainty predictions, we perform two experiments. First, we compare the calibration (Leibig et al., 2017) of dropout uncertainty to predicted uncertainty removing, in 10% increments between 0% and 50%, the most uncertain predictions and re-evaluating accuracy. Well-calibrated uncertainty should result in an increase in accuracy as uncertain predictions are removed.

Second, we compare the rankings of data points based on the dropout uncertainty and predicted uncertainty. Let $x_{\text{predicted}}$ be the index of data point x when data points are sorted by predicted uncertainty, x_{dropout} be the index of x when data points are sorted by dropout uncertainty. Then, we calculate average normalized distance as the average of $\frac{|x_{\text{predicted}} x_{\text{dropout}}|}{n}$ for each x in the test set, where n is the number of data points. We also report the standard deviations the aforementioned values.

5 Results

5.1 Regression Metrics

Tables 2 - 4 present traditional regression metrics for estimating uncertainty on unseen evaluation data. Average R^2 is the average value of the R^2 correlation coefficient between true uncertainty and predicted uncertainty generated from our technique. We also report mean absolute error between true uncertainty and predicted uncertainty generated from our technique.

Table 2: Uncertainty regression results for Adult Income data

Regression Algorithm	**Uncertainty Output Type**	**Average R^2**	**Mean Absolute Error**
Decision Tree	Multi-Single	0.873	0.021
Decision Tree	Multi-Output	0.875	0.021
Neural Network	Multi-Single	0.964	0.012
Neural Network	Multi-Output	0.966	0.011
XGBoost	Multi-Single	0.949	0.014

Table 3: Uncertainty regression results for Trauma Triage data

Regression Algorithm	**Uncertainty Output Type**	**Average R^2**	**Mean Absolute Error**
Decision Tree	Multi-Single	0.875	0.017
Decision Tree	Multi-Output	0.880	0.017
Neural Network	Multi-Single	0.961	0.011
Neural Network	Multi-Output	0.963	0.010
XGBoost	Multi-Single	0.955	0.011

Table 4: Uncertainty regression results for MNIST data (* represents negative correlation)

Regression Algorithm	**Uncertainty Output Type**	**Average R^2**	**Mean Absolute Error**
Decision Tree	Multi-Single	0.283*	0.044
Decision Tree	Multi-Output	0.281*	0.044
Neural Network	Multi-Single	0.730	0.022
Neural Network	Multi-Output	0.665	0.022
XGBoost	Multi-Single	0.888	0.014

5.2 Uncertainty Calibration

Figure 1 presents the uncertainty calibration results for each dataset, estimated uncertainty, and dropout uncertainty. These plots show the change in accuracy as the most uncertain predictions are removed from accuracy calculations.

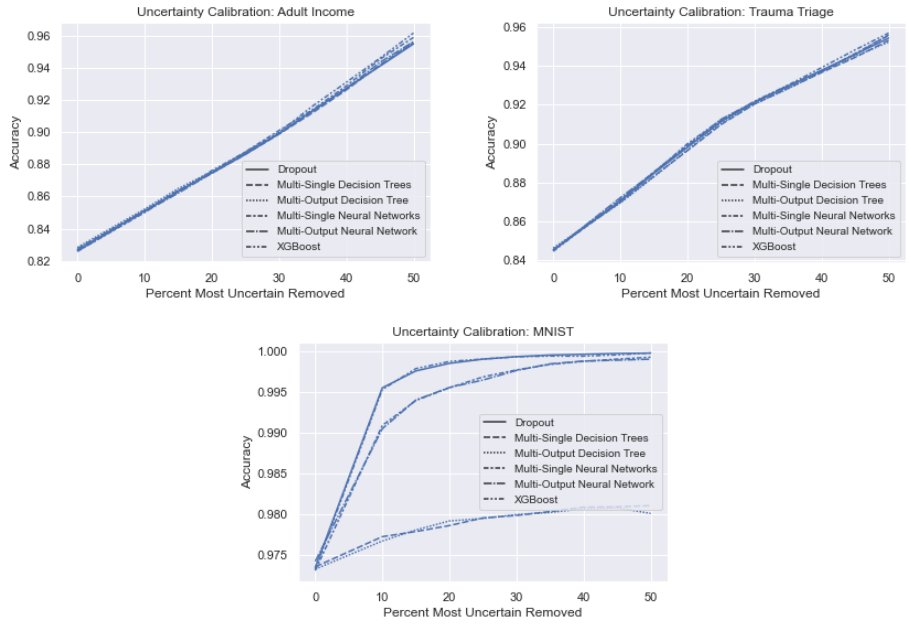

Figure 1: Uncertainty calibration of proposed technique compared to dropout uncertainty

5.3 Uncertainty Ordering Correlation

Tables 5 - 7 present the results comparing the locations of each element when the data points are sorted by predicted uncertainty measure compared to dropout uncertainty.

6 Discussion

6.1 Regression Metrics

Tables 2 - 4 present the R^2 correlation coefficient for our inferred uncertainty compared to the "true" dropout uncertainty. We find for Adult Income and Trauma Triage, regression yields an R^2 value of at least 0.87 with values peaking at 0.95. This implies for Adult Income and Trauma Triage, regression techniques have a close goodness-of-fit to dropout uncertainty. Moreover, mean absolute errors for these techniques are less that 0.2 for all regression techniques for both datasets.

Table 5: Uncertainty ordering correlation results for Adult Income data

Regression Algorithm	Uncertainty Output Type	Average Normalized Distance	Normalized Standard Deviation
Decision Tree	Mulit-Single	0.077	0.071
Decision Tree	Multi-Output	0.076	0.069
Neural Network	Multi-Single	0.043	0.038
Neural Network	Multi-Output	0.042	0.036
XGBoost	Multi-Single	0.048	0.040

Table 6: Uncertainty ordering correlation results for Trauma Triage data

Regression Algorithm	Uncertainty Output Type	Average Normalized Distance	Normalized Standard Deviation
Decision Tree	Multi-Single	0.076	0.074
Decision Tree	Multi-Output	0.075	0.072
Neural Network	Multi-Single	0.043	0.037
Neural Network	Multi-Output	0.042	0.036
XGBoost	Multi-Single	0.052	0.048

For MNIST (see Table 4), we find that both decision tree models do not estimate uncertainty well, having an average R^2 less than zero, implying that whatever relationship exists between the estimated uncertainty and dropout uncertainty is minimal and an inverse relationship. Thus, we can conclude decision trees do not effectively estimate uncertainty for MNIST.

Fortunately, however, this is not true for the neural network models or the gradient-boosting trees (XGBoost). The neural networks achieve average R^2 values of at least 0.665, and XGBoost have an average R^2 of 0.888. These more complex models are successful at using both the increased dimensionality and classification prediction information to infer epistemic uncertainty.

We suspect that this difference in performance between the tabular datasets (Adult Income and Trauma Triage) and MNIST is a result of the datasets' dimensionality. MNIST is an image dataset with 784 input pixels. Since 10 classes can be predicted from MNIST data, this results in 794 input features into the epistemic uncertainty inference models. Future work extending this technique to other types of data (e.g., natural language, images, etc.) may benefit from a multimodal approach in the inference network (Brown et al., 2020).

6.2 Uncertainty Calibration

Our next experiment evaluates how well an uncertainty measure can identify potentially incorrect predictions. Several works in the existing literature (Gal & Ghahramani, 2016; Leibig et al., 2017; Brown & Talbert, 2019) demonstrate the effectiveness of dropout uncertainty quantification at identifying likely incorrect instances and the benefit of using such a technique in medicine.

Figure 1 demonstrates how accuracy changes as the most uncertain data points are removed for each dataset. For Adult Income and Trauma Triage, each regression technique results in similar changes in accuracy as more uncertain data points are removed. For MNIST, neither of the decision tree regressors are able to achieve the same change in accuracy as dropout or other uncertainty estimation techniques. This is to be expected, as decision tree regressors did not exhibit high R^2 values. Neural networks and XGBoost, however, more closely mimic dropout calibration with XGBoost outperforming both neural network regressors. This is also supported by R^2 coefficient values in Table 4 which indicated that XGBoost had a better goodness-of-fit compared to neural networks.

Table 7: Uncertainty ordering correlation results for MNIST data

Regression Algorithm	Uncertainty Output Type	Average Normalized Distance	Normalized Standard Deviation
Decision Tree	Multi-Single	0.223	0.203
Decision Tree	Multi-Output	0.233	0.212
Neural Network	Multi-Single	0.134	0.121
Neural Network	Multi-Output	0.147	0.130
XGBoost	Multi-Single	0.106	0.108

6.3 Uncertainty Ordering Correlation

Finally, we compare the ordering of data points when the datasets are sorted by the dropout uncertainty and their estimated uncertainty. Tables 5 - 7 present the average distance and normalized standard deviations of the indices of each data point normalized by the total number of elements. We find that for Adult Income (Figure 5) and Trauma Triage (Figure 6), on average, data points are within 10% of each other for each regression technique. For MNIST (Figure 7), we find that indices are within 15% for more accurate regression techniques (neural networks and XGBoost) and within 25% for decision trees.

6.4 Efficiency of Proposed Technique

As mentioned previously, this technique is motivated by a need to produce epistemic uncertainty values without extensive modifications to neural network architectures or performing multiple inferences on pre-existing, expensive neural network architectures. This technique reduces the number of inferences necessary to be between 1 and the total number of classes.

We evaluated using a single neural network or decision tree that produces multiple regression outputs with one inference through the model. In general, we find for tabular data, there is little difference in uncertainty estimations. Estimations produced by a single model with multiple outputs tended to be marginally less effective than single regression models per class. However, for high-dimensionality data, such as MNIST, we find more substantial shifts in efficacy between using multiple single-output models compared to a single model with multiple output.

In production, the number of inferences necessary for uncertainty estimations could be reduced to one by producing the classification through the classification neural network and only utilizing the uncertainty estimator associated with the most probable class. Thus, although this technique will require training and tuning more regression models (up to the number of classes to predict), it reduces computation power and resources necessary in production.

7 Conclusion

In this work, we present a technique that allows epistemic uncertainty to be estimated through regression-based machine learning algorithms. This technique requires fewer machine learning inferences to yield uncertainty estimates comparable to classic, but resource-expensive, techniques such as dropout.

This work has several avenues through which it can be extended. First, we find that simpler, less-expensive machine learning techniques perform poorly on high-dimensional data such as images. Future work could explore using multi-modal machine learning and other techniques to effectively infer epistemic uncertainty values. Moreover, this technique can also be extended to other measures of uncertainty such as aleatoric uncertainty, to provide a more complete picture of the uncertainty associated with a prediction. Finally, we would like to explore how XAI could be applied to epistemic uncertainty estimators to determine if it can provide insight into causes of epistemic uncertainty.

ETHICS STATEMENT

The Trauma Triage Dataset was retrieved from a Level 1 Trauma Center. Use of the trauma triage data received internal review board approval.

REPRODUCIBILITY STATEMENT

Section 4 (Experimental Methodology) includes the following information to aid in reproducibility of results:

- Names of code libraries for machine learning and deep learning algorithms
- Non-default parameters for each machine learning and deep learning algorithm used for both classification and regression
- Neural network architectures, loss functions, optimization algorithms (with default parameters), number of epochs, and regularization techniques used for both classification and regression
- Data preprocessing steps for categorical and continuous data

Source code for this paper can be downloaded from `https://github.com/somebody-anonymous/iclr_2022_code/archive/refs/heads/main.zip`

REFERENCES

Martín Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The Need for Uncertainty Quantification in Machine-Assisted Medical Decision Making. *Nature Machine Intelligence*, 1(1):20–23, 2019.

James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, pp. 35. New York, NY, USA., 2007.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

Katherine E Brown and Douglas A Talbert. Estimating uncertainty in deep image classification. In *AMIA*, 2019.

Katherine E Brown, Farzana Ahamed Bhuiyan, and Douglas A Talbert. Uncertainty quantification in multimodal ensembles of deep learners. In *The Thirty-Third International Flairs Conference*, 2020.

Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug 2016. doi: 10.1145/2939672.2939785. URL `http://dx.doi.org/10.1145/2939672.2939785`.

François Chollet et al. Keras, 2015. URL `https://keras.io`.

Elaine Cole, Fiona Lecky, Anita West, Neil Smith, Karim Brohi, and Ross Davenport. The impact of a pan-regional inclusive trauma system on quality of care. *Annals of Surgery*, 264(1):188–194, 2016.

Gregory F Cooper et al. An Evaluation of Machine-Learning Methods for Predicting Pneumonia Mortality. *Artificial Intelligence in Medicine*, 9(2):107–138, 1997.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017. URL http://archive.ics.uci.edu/ml.

Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4): 367–378, 2002.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1050–1059, 2016.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Ramyad Hadidi, Jiashen Cao, Yilun Xie, Bahar Asgari, Tushar Krishna, and Hyesoon Kim. Characterizing the deployment of deep neural networks on commercial edge devices. In *2019 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 35–48. IEEE, 2019.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural Networks for Machine Learning: Overview of mini-batch gradient descent, 2012.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pp. 5574–5584, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, 2014.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1):1–14, 2017.

Ji Li, Zihao Yuan, Zhe Li, Ao Ren, Caiwen Ding, Jeffrey Draper, Shahin Nazarian, Qinru Qiu, Bo Yuan, and Yanzhi Wang. Normalization and dropout for stochastic computing-based deep convolutional neural networks. *Integration*, 65:395–403, 2019.

Nehemiah T Liu and Jose Salinas. Machine learning for predicting outcomes in trauma. *Shock: Injury, Inflammation, and Sepsis: Laboratory and Clinical Approaches*, 48(5):504–510, 2017.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, pp. 3, 2013.

T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN 9780071154673. URL https://books.google.com/books?id=EoYBngEACAAJ.

Tomoyuki Myojin, Shintaro Hashimoto, and Naoki Ishihama. Detecting uncertain bnn outputs on fpga using monte carlo dropout sampling. In *International Conference on Artificial Neural Networks*, pp. 27–38. Springer, 2020.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

F. Pedregosa et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pp. 63–71. Springer, 2003.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Scott M Sasser et al. Guidelines for Field Triage of Injured Patients: Recommendations of the National Expert Panel on Field Triage, 2011. *Morbidity and Mortality Weekly Report: Recommendations and Reports*, 61(1):1–20, 2012.

Sota Sawaguchi and Hiroaki Nishi. Slightly-slacked dropout for improving neural network learning on fpga. *ICT Express*, 4(2):75–80, 2018.

Donald F Specht. A general regression neural network. *IEEE transactions on neural networks*, 2 (6):568–576, 1991.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Christopher KI Williams. Computing with infinite networks. In *Advances in neural information processing systems*, pp. 295–301, 1997.

Shenyuan Xu, Size Liu, Hua Wang, Wenjie Chen, Fan Zhang, and Zhu Xiao. A hyperspectral image classification approach based on feature fusion and multi-layered gradient boosting decision trees. *Entropy*, 23(1):20, 2021.

A Appendix

Tables 8, 9, 10 present the accuracy of each technique as most uncertain predictions are removed from accuracy calculations.

Table 8: Uncertainty calibration results for Adult Income data

Regression Algorithm	**Uncertainty Output Type**	**Percent Removed**					
		0%	**10%**	**20%**	**30%**	**40%**	**50%**
Dropout	N/A	0.827	0.851	0.875	0.899	0.927	0.955
Decision Tree	Multi-Single	0.828	0.851	0.874	0.899	0.926	0.956
Decision Tree	Multi-Output	0.828	0.852	0.875	0.899	0.927	0.956
Neural Network	Multi-Single	0.826	0.850	0.875	0.901	0.929	0.959
Neural Network	Multi-Output	0.826	0.850	0.875	0.899	0.927	0.956
XGBoost	Multi-Single	0.826	0.851	0.876	0.90	0.931	0.961

Table 9: Uncertainty calibration results for Trauma Triage data

Regression Algorithm	**Uncertainty Output Type**	**Percent Removed**					
		0%	**10%**	**20%**	**30%**	**40%**	**50%**
Dropout	N/A	0.845	0.870	0.898	0.921	0.938	0.954
Decision Tree	Multi-Single	0.845	0.870	0.896	0.920	0.937	0.953
Decision Tree	Multi-Output	0.846	0.870	0.897	0.920	0.937	0.952
Neural Network	Multi-Single	0.845	0.871	0.899	0.921	0.937	0.956
Neural Network	Multi-Output	0.845	0.872	0.899	0.921	0.938	0.956
XGBoost	Multi-Single	0.845	0.871	0.900	0.922	0.939	0.957

Table 10: Uncertainty calibration results for MNIST data

Regression Algorithm	**Uncertainty Output Type**	**Percent Removed**					
		0%	**10%**	**20%**	**30%**	**40%**	**50%**
Dropout	N/A	0.974	0.995	0.998	0.999	1.000	1.000
Decision Tree	Multi-Single	0.974	0.977	0.979	0.980	0.981	0.981
Decision Tree	Multi-Output	0.973	0.977	0.979	0.980	0.981	0.980
Neural Network	Multi-Single	0.973	0.991	0.996	0.998	0.999	0.999
Neural Network	Multi-Output	0.974	0.991	0.996	0.998	0.999	0.999
XGBoost	Multi-Single	0.973	0.995	0.999	0.999	0.999	1.000