UNCERTAINTY ESTIMATION IN NEURAL NETWORKS THROUGH MULTI-TASK LEARNING

Ashish James and Anusha James

Insitute for Infocomm Research (I2R), Agency for Science Technology and Research (A-STAR), Singapore

ABSTRACT

The impressive predictive performance of deep learning techniques on a wide range of tasks has led to its widespread use. Estimating the confidence of these predictions is paramount for improving the safety and reliability of such systems. However, the uncertainty estimates provided by neural networks (NNs) tend to be overconfident and unreasonable. Previous studies have found out that ensemble of NNs typically produce good predictions and uncertainty estimates. Inspired by these, this paper presents a new framework that can quantitatively estimate the uncertainties by leveraging the advances in multi-task learning through slight modification to the existing training pipelines. This promising algorithm is developed with an intention of deployment in real world problems which already boast a good predictive performance by reusing those pretrained models. The idea is to capture the behavior of the trained NNs for the base task by augmenting it with the uncertainty estimates from a supplementary network. A series of experiments show that the proposed approach produces well calibrated uncertainty estimates with high quality predictions.

KEYWORDS

Uncertainty estimation, Neural Networks, Multi-task Learning, Regression.

1. INTRODUCTION

The significant advancements and tremendous performance of neural networks (NNs) has led to its ubiquitous usage in many fields. Even with such leading edge deep learning techniques, the interpretability and quantification of predictive uncertainty in NNs has long been a challenging and yet unsolved problem [1, 2, 3]. However, this is crucial for integrating deep learning algorithms into most practical applications as well as supporting other applications such as to initiate exploration/exploitation in reinforcement learning (RL), active learning etc. Despite NNs boasting state-of-the-art accuracy in supervised learning benchmarks, they are poor at quantifying predictive uncertainty, and tend to produce overconfident and miscalibrated predictions. This is significantly useful in algorithms which helps to take critical decisions that impact human lives in medical, financial or legal domains, where the cost of error is high. For instance, in safety-critical applications like autonomous driving system need to respond based on the confidence of the AI system. Hence, it is imperative to estimate the uncertainty in model’s predictions as it enhances the reliability of AI systems and trigger actions in an informed manner.

While more data and better training and optimization techniques can improve the predictive performance of NNs, it is crucial for these networks to emphasize the uncertainty of predictions to be trustworthy. The NNs predict presumptuously because they do not specifically model data uncertainty or they overlook relationship between data and model uncertainty [4]. Uncertainty usually originates from two sources : data uncertainty (epistemic) and model uncertainty

DOI:10.5121/ijaia.2023.14402

(aleatoric). For example a sensor can always be expected to give noisy data and training datasets need not always cover all the possible edge-cases. Uncertainty estimation can be used in several scenarios such as determination of samples which are hard to classify, evaluation of uncertainty on out-of-distribution samples. These algorithms should predict a rare/larger noisy sample with higher uncertainty, when compared to a frequent/less noisy sample in the training data.

Traditional uncertainty estimation algorithms model the network activations and weights by parametric probability distributions, which are difficult to train [5]. Recently, several approaches have explored incorporating uncertainty and probabilistic methods to NNs [1, 3, 5, 6]. Majority of these works exploits the Bayesian framework [7, 8, 9, 10] as a principled approach for modeling uncertainty. In such networks, the parameters of the NNs are modeled as probability distributions computed via the Bayes rule [11]. Since it is computationally intractable for exact Bayesian inference in NNs, several approximations have been developed [11, 12, 13, 8, 14, 15, 16, 17, 18, 19]. Whilst appealing, the quality of uncertainty estimates by these networks depends on the degree of approximations, and the assumptions made on the prior distribution [20]. Further, such networks are often harder to implement and computationally expensive compared to widely adopted non-Bayesian NNs.

In this context, estimating uncertainty through non-Bayesian NNs using a generalized approach that can deliver high quality uncertainty estimates through slight modifications is required. Monte Carlo dropout (MC-dropout) has been recently proposed along these lines by estimating the predictive uncertainty using dropout during both training and inference times [1]. This technique has gained popularity in practice owing to its easy implementation and slightest modification required to the existing pipelines. Interestingly, such an approach can be interpreted as an ensemble of models [21], wherein the dropout can be considered to perform averaging across the possible subnetworks. Along these lines, a technique that utilizes an ensemble of models trained with a proper scoring rule to produce accurate uncertainty estimates has been proposed [3].

In this work, we propose an appealing technique which exploits the multi-task learning framework for quantifying uncertainty. First a base network is trained on the dataset for the specific task, and then the learned features are exploited for providing the uncertainty estimates as an addendum using another network. Intuitively, the idea is to preserve the source knowledge acquired by the important neurons for the original task. Such an approach permits the neurons to retain their abilities to extract features relevant for the original task, and augment it for the uncertainty estimation. Our main contribution are as follows:

- A simple and extensible multi-task learning framework that can achieve state-of-the-art performance for supervised learning on many publicly available datasets while achieving good uncertainty estimates.
- Easy extension of the widely adopted deep neural networks that are trained for the original task to estimate uncertainties as an addendum. This is achieved by adding an uncertainty estimating network to existing models from standard pipelines and retrained using proper loss function for computing the uncertainty.

These properties make this an appealing approach for uncertainty estimation, enabling them to be easily incorporated into many potential applications, to improve the reliability of AI applications.

2. RELATED WORK

Estimating the quality of predictive uncertainties is challenging because of the lack of ground truth uncertainties. However, there has been lot of recent interest to capture uncertainty of AI systems due to its wide scale adoption and most of these can be broadly classified into Bayesian and non-Bayesian techniques.

2.1. Bayesian Approaches

These approaches compute uncertainty by specifying a distribution over model parameters and then, given the training data, these parameters are marginalized to form a posterior predictive distribution [13, 8, 10]. However, for modern neural networks that contain millions of parameters, the posterior over these parameters (loss surface) is highly non-convex, rendering them infeasible in these scenarios. Hence several modern approaches to Bayesian deep learning are based on a variety of approximations such as Laplace [12], Markov chain Monte Carlo (MCMC) [13], expectation propagation [16, 22], variational inference [8, 15, 14] etc. Laplace approximations assume a Gaussian posterior and has been utilized for Bayesian neural networks in [12]. Hamiltonian Monte Carlo (HMC) approach for inference in NNs is proposed in [13]. In practice, the approximation introduces errors in modern NNs with finite learning rates and also difficulty with tuning stochastic gradient MCMC methods has led to the exploration of other techniques.

A scalable approximate Bayesian inference technique for deep learning is proposed in [23], where the information contained in the SGD trajectory is utilized to efficiently approximate the posterior distribution of the weights of the NNs. [14] discusses a Gaussian variational posterior approximation over the weights of NNs, which is generalized in [24] for the reparameterization trick for training deep latent variable models. Even though variational techniques achieve good performance for moderately sized networks, they are empirically found to be difficult to train on larger architectures [25]. A probabilistic back propagation (PBP) technique for learning Bayesian NNs is proposed in [5], where a forward propagation of probabilities through the network and backward propagation of gradients are computed.

However, the quality of predictive uncertainties obtained using such Bayesian approaches crucially depends on (i) the degree of approximations employed due to computational constraints and (ii) correctness of prior distribution assumption as convenient priors can lead to unreasonable predictive uncertainties [3].

2.2. Non-Bayesian Approaches

The computational demands required by Bayesian approaches for uncertainty estimation has led researchers to work on exploring other methods with less computational and time cost. One such recent approach which is computationally much efficient, the Monte Carlo dropout (MC-dropout) method, has been shown to be an approximation of the probabilistic Bayesian model [1]. Concisely, the method performs multiple stochastic forward passes by activating the dropout during inference times as well to provide a Monte Carlo sampling utility that could reflect uncertainty estimates. However, such an approach is applicable only for dropout models and in practice it is harder to find an optimal dropout rate that can achieve good predictive performance and uncertainties.

Dropouts can be viewed as a form of ensemble learning, where ensembling is done over an exponential number of subnetworks sampled from the original network [21]. Recent studies have shown that ensemble approaches are not only effective at improving predictive accuracy but provides good model uncertainty estimates as well [3, 26, 27, 28]. In most of such studies, an ensemble of models with different weight initializations are constructed with mean prediction and variance as the uncertainty estimate. [6] discusses a method of estimating both the uncertainty and prediction using a single NN. An ensembling technique of NNs using a proper scoring rule that can provide both aleatoric and epistemic uncertainty is shown in [3]. Such a non-Bayesian approach has been found to outperform MCdropout with significantly less samples. This can be attributed to the increased capacity of NNs used in deep ensembles as it does not require dropout at inference times and also the different weight initializations causes the network to converge to different minimas [29].

One of the main drawbacks with the above approaches is that it requires significant modification to the existing pipelines. Further, over the years, numerous highly accurate state-of-the-art models has been trained which frightfully lack the ability to estimate uncertainty, which is paramount for real world applications. Also, when we need to perform a new task of uncertainty estimation, these existing models cannot be utilized. Hence we propose this minimally invasive framework leveraging multi-task learning, to circumnavigate the above concerns while estimating the uncertainty. This works as a supporting tool that can augment any pretrained NN with uncertainty estimates, by fine-tuning a trained model using a loss criterion that enables to obtain predictions along with uncertainty. Further details of the proposed approach is provided in the next section.

3. Uncertainty Estimation Through Multi-Task Learning

This section provides detailed summary of the proposed framework which combines the predictive capability of a pretrained network and uncertainty estimation from a supplementary network. A block diagram of the proposed framework is shown in Figure 1, which depicts the pretraining followed by fine tuning an ensemble of NNs, to capture uncertainties along with the predictions.

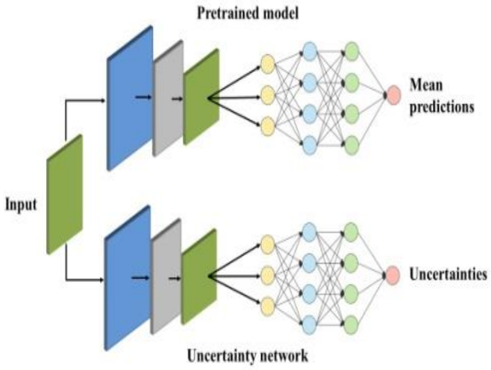


Figure 1. Block diagram of the proposed approach by integrating pretrained and uncertainty subnet works to obtain predictions and its associated uncertainties

3.1. Problem Setup

It is assumed that the training dataset D consists of N i.i.d data points $D = \{x_n, y_n\}_{n=1}^{N}$, where $x \in R^I$ represents the I-dimensional input features and y represents the corresponding output features. The y labels for classification problems are considered to be one among the K classes, and for regression $y \in R^O$ represents the O-dimensional real valued output features. The problem we are trying to tackle is to obtain good quality predictions along with its associated uncertainty estimates. The proposed approach tries to disentangle this by initially training an ensemble of NNs to model the predictive distribution $p_\theta(y|x)$, where θ are the parameters of the NNs, followed by fine-tuning these NNs along with the supplementary uncertainty network for estimating the associated uncertainties. There are two types of uncertainties which need to be considered for deep learning:

- Aleatoric uncertainty captures the inherent noise in data, usually due to the imperfect sensors. This type of uncertainty is irreducible.
- Epistemic uncertainty captures the ignorance of the model. This occurs due to an imbalance in the training data distribution. In contrast to aleatoric uncertainty, this uncertainty can be reduced by collecting more training samples in regions where the models are uncertain.

In the proposed approach, both these types of uncertainties can be estimated.

3.2. Pretraining of Forward Models

For the above supervised problem formulation, the design objective of pretraining is to discover the mapping from inputs x to outputs y. For a particular learning task, the weight parameters of the NNs are updated during the training phase so that the model generalizes well for the training data. This is achieved by using a loss/cost function that quantifies the closeness of predictions with the real outputs. In other words, training phase of NNs aim to find the ideal weights and biases which minimizes the error between actual and predicted outcomes. The proposed approach provides the flexibility of using an existing (pretrained) model trained for the base learning task and leveraging it for the uncertainty estimation task.

Further, the proposed approach exploits ensemble learning where a group of NNs are constructed and their predictions are combined to provide improved generalization. This is achieved by aggregating multiple diverse versions of NNs (as they have converged to different local minima and make different prediction errors) learned for a specific task which enables it to perform better than any of the individual models. In this paper, the individual NNs that constitute the ensemble is trained on the entire dataset and differ only in their initialization. Such an approach has been observed to yield better predictive accuracy and uncertainty estimates [3, 30]. For simplicity, the ensemble is treated as a uniform mixture model and the predictions are combined as $p(y|x) = \frac{1}{M}\sum_{m=1}^{M} p_{\theta_m}(y|x)$, where M corresponds to the ensemble size. Once a good predictive model is obtained, the next phase is to integrate uncertainty estimation into this framework and is explained in the following section.

3.3. Multi-Task Learning from Pretrained Forward Models to Estimate Uncertainty

This phase is motivated by scenarios where good forward models exist and additional uncertainty measure will enhance its usability and trustability. We believe the concept of multi-task learning would be suitable for a minimally invasive approach for uncertainty estimation with good potential, since the features are general and suitable for both the base and uncertainty task. Hence, multi-task learning is exploited in this context, where a base learner is adapted to improve the generalization of a new task. As shown in Figure 1, the pretrained networks (or already trained models) is augmented with supplementary uncertainty estimation subnetworks to estimate uncertainty. Concisely, the base learner for the prediction task augments the learning of the target conditional probability distribution $p_\theta\,(y|x)$ through the multi-task of knowledge from the base task that has been already learned. This is achieved by integrating a separate uncertainty estimation network with the base learner and trained with a proper loss function based on the supervised problem setting.

For example in a regression setting with O = 1, the base learner outputs the mean prediction $\mu(x)$ and the uncertainty estimation network outputs the variance $\sigma^2(x)$. This variance $\sigma^2(x)$ is modelled by a separate uncertainty estimation network with an exponential activation function to ensure that variance is always positive. For simplicity, y is considered sampled from a Gaussian distribution with the mean prediction and variance, the loss function that need to be minimized is the negative log likelihood (NLL) given by [3, 6]

$$-\log p_\theta(y_n|x_n) = \frac{\log \sigma^2_{\theta_u}(x)}{2} + \frac{(y - \mu_{\theta_b}(x))^2}{2\sigma^2_{\theta_u}(x)} \qquad (1)$$

where θ_b and θ_u corresponds to the parameters of the base and uncertainty estimation network, respectively.

This training process inherently fine-tunes the weights of the pretrained NNs and adjusts the weights of the uncertainty network for estimating the variance of the target error distribution as a function of input. All the NNs that constitute the ensemble is then trained accordingly and the overall training procedure is summarized in the pseudo code below.

By considering the ensemble as a uniform mixture model, the predictions are aggregated from the individual models. In a supervised classification setting, this corresponds to averaging the predicted probabilities and for regression, this becomes a mixture of Gaussian distributions [3]. In a regression setting, the mean and variance of a Gaussian mixture is given as

$$\mu_e(x) = \frac{1}{M}\sum_{m=1}^{M}\mu^i_{\theta_b}(x) \qquad (2)$$

$$\sigma_e(x) = \underbrace{\frac{1}{M}\sum_{m=1}^{M}\sigma^i_{\theta_u}(x)^2}_{Aleatoric} + \underbrace{\frac{1}{M}\sum_{m=1}^{M}\mu^i_{\theta_b}(x)^2 - \mu_e(x)^2}_{Epistemic} \qquad (3)$$

where $\mu_e(x)$ and $\sigma_e(x)$, corresponds to the ensemble mean and variance (uncertainty), respectively. The variance term in equation (3) combines both the aleatoric and epistemic uncertainty estimates.

4. EXPERIMENTS

In this paper, NLL is employed as the loss criterion, which is also a commonly used evaluation metric for predictive uncertainty estimation [31]. For supervised regression problems, the root mean square error (RMSE) is also evaluated to highlight

Algorithm 1: Pseudocode of the proposed approach

```
1. Construct an ensemble of base networks to model the predictive distribution $p_\theta(y|x)$.
   Initialize $\theta_{b1}, \theta_{b2}, \cdots, \theta_{bm}$ randomly. Recommended ensemble size is 5.
2. for $m = 1 : M$ do
      (a) Sample data points randomly for each NN, so that each member of the data has an equal opportunity of being picked.
      (b) Train the base networks independently, until cost function for the base task is minimized.
   end
3. Construct an ensemble of multi-task networks for the uncertainty task.
   This is done by copying the base network and randomly initializing another uncertainty network for computing the uncertainty. A proper loss criterion $L(\theta,x,y)$ which captures the predictive distribution with uncertainty estimates is utilized.
4. for $m = 1 : M$ do
      (a) Fine-tune the base NNs along with the uncertainty networks until the loss function is minimized.
   end
```

that uncertainty estimation doesn't impact the prediction capability of the proposed approach.

4.1. Synthetic Regression Problem

Firstly, the performance of the proposed approach is evaluated on a one dimensional synthetic regression problem. Experimenting with a synthetic regression problem enables us to control the target distribution y with a known noise distribution which will validate our proposed method of predicting the mean and its associated uncertainties both aleatoric and epistemic. The aleatoric component should capture the noise variance in the data and epistemic component should capture the model uncertainties where there is no/less training data. Similar to [6], an amplitude modulation of the input feature for the target distribution is considered and is given as

$$f(x) = \sin(4x) \sin(5x) \tag{4}$$

The output feature y is obtained by corrupting the target distribution with noise and is modeled as

$$y = f(x) + N(x) \tag{5}$$

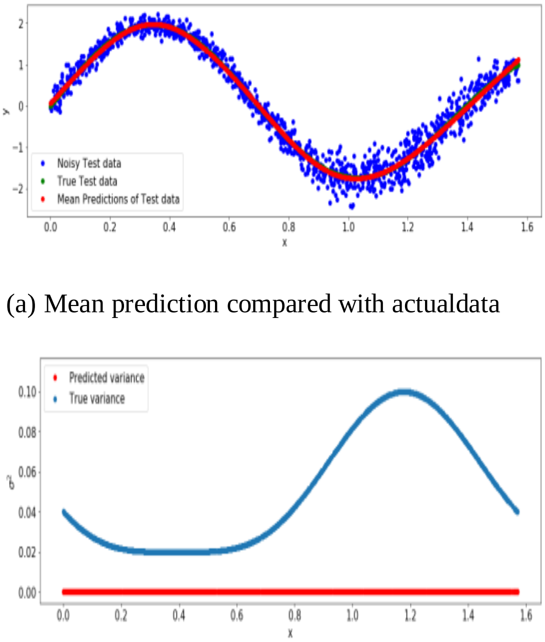
The additive noise $N(x)$ is considered to be normally distributed with variance $\sigma^2(x)$, which moves the targets away from their true values $f(x)$. For the synthetic regression problem, the variance of the noise is considered as

$$\sigma^2(x) = 0.02 + 0.02(1 - \sin(4x))^2 \tag{6}$$

Around 5000 samples are generated based on equation (5) over the interval $x \in [0, \pi/2]$ and around 75% are allocated for training and remaining 25% for testing. In the proposed approach, the base task is to estimate the true mean (μ) of the noisy target distribution around $f(x)$. The initial weights of the base network are based on the default weight initialization in Tensor flow and mean squared error (MSE) is used as the loss criterion which enables the NNs to map the target distribution closely. Once the training of the base network has reached a point where loss criterion is improving only very slowly, we do an early stopping, as additional training in the next phase can fine-tune the predictions.

Figure 2 illustrates the performance of the pretrained network on the test set. It can be observed from Figure 2a that the pretrained ensemble network closely approximates the true function (overlapping green and red line). However, the uncertainty measured as the variance among the NNs constituting the ensemble (red line) does not accurately model the noise inherent in the data (blue line) as shown in Figure 2b.

In order to improve the performance of the uncertainty estimation task, the neural networks for the base task is copied and concatenated with a randomly initialized uncertainty network that estimates the true variance $\sigma^2(x)$ of the target error distribution, which is actually the quantitative uncertainty due to $N(x)$. The noise $N(x)$ can either be independent of the input or systematically vary over the



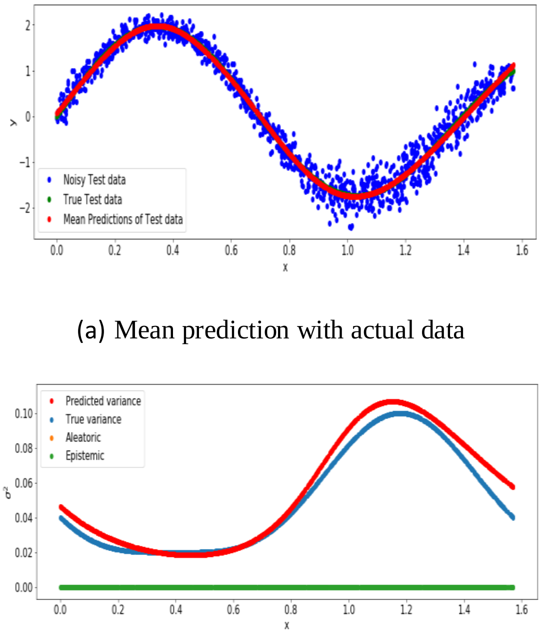
(a) Mean prediction compared with actualdata

(b) Variance prediction compared with actual variance

Figure 2: Comparison of predicted mean and variance from the pretrained ensemble of networks to the target probability distribution with the actual mean and variance

input space. As mentioned previously, here we apply our method on a synthetic example with a noise that varies with the input feature. This network is then trained to minimize the NLL cost function (equation (1)), which reflects both the accuracy and quality of predictive uncertainty. The outputs of the final network gives us the parameters of a normal distribution, which describes the data generated by the sampling process in equation (5). The results of the proposed approach on the test set are shown in Figure 3. Figure 3a shows the training data x (over the interval $x \in [0,\pi/2]$), the true function $f(x)$, and the mean prediction by the network. It clearly shows that the predictions ($\mu_e(x)$) from the networks trained using the proposed framework closely matches the true function $f(x)$ (overlapping green and red lines). Further, Figure 3b compares the true variance $\sigma^2(x)$ with the predicted variance $\sigma_e(x)$ which has both aleatoric and epistemic uncertainty components as described in equation (3). It clearly shows that proposed framework maps the true noise variance in the data accurately. Hence, this validates that the proposed approach accurately estimates the mean and variance of the target probability distribution. Further in the interpolation region, where training data is available, the data noise (aleatoric uncertainty) is the main uncertainty contributor as observed by overlapping predicted noise variance and aleatoric component in Figure 3b. In this region, the contribution from model uncertainty (epistemic) should be quite small which is also validated in Figure 3b.

After validating the performance of the proposed approach in the interpolation region, the uncertainties are estimated in the extrapolation region by varying x over the interval $[-\pi/2, 0]$. The performance of the proposed approach with mean prediction and uncertainties shown by shaded region corresponding to three standard deviations are shown in Figure 4. It can be observed that in the interpolation region, the data lies within the uncertainty estimated by the proposed approach. Also, the uncertainty starts to increase as we move further away from the interpolation region into the extrapolation region. This validates that the proposed approach estimates higher uncertainties in regions where there is no/less training data.



(a) Mean prediction with actual data

(b) Variance prediction with actual variance

Figure 3: Comparison of predicted mean and variance from the proposed framework of the target probability distribution with the actual mean and variance

The results shown in Figures 3, 4 validates that (i) the proposed approach is able to provide a good estimate of both mean and uncertainty in the interpolation region and (ii) uncertainty increases as we move farther from the observed training data into the extrapolation region. These are important aspects that enhances the value of the proposed approach.

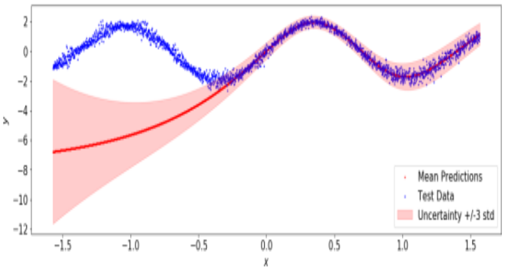


Figure 4: Mean prediction and uncertainty estimated by the proposed approach in the interpolation and extrapolation region

4.2. Regression on Real World Datasets

Following the validation on synthetic regression problem, we empirically evaluated the performance of our method and compared to high performing baselines like deep ensembles [3], probabilistic back propagation (PBP) [5], MC- Dropout [1], and Bayesian linear regression ensemble (BLR) [32]. The entire training set is used to train each network in the ensemble, since NNs typically perform better with more data. We use the experimental setup proposed in [5], which is also adopted in [1] and [3] to evaluate MC-dropout and deep ensembles, respectively.

We use a 1-hidden layer NN with 50 hidden units for smaller datasets and 100 hidden units for larger datasets. The hidden layers are given rectified linear unit (ReLU) activation in both the base and uncertainty networks, and output layers

Table 1: Results on regression real-world datasets comparing RMSE and NLL performance metrics

			RMSE				NLL			
Datasets	N	I	PBP	MC-dropout	Deep Ensembles	Proposed Approach	PBP	MC-dropout	Deep Ensembles	Proposed Approach
Boston housing	506	13	3.01± 0.18	2.97 ± 0.85	3.28 ± 1.00	2.96 ± 0.16	2.57± 0.09	2.46 ± 0.25	2.41 ± 0.25	2.01 ± 0.32
Concrete strength	1030	8	5.67± 0.09	5.23 ± 0.53	6.03 ± 0.58	5.03 ± 0.76	3.16± 0.02	3.04 ± 0.09	3.06 ± 0.18	2.82 ± 0.62
Energy efficiency	768	8	1.80± 0.05	1.66 ± 0.19	2.09 ± 0.29	2.03 ± 0.22	2.04± 0.02	1.99 ± 0.09	1.38 ± 0.22	1.11 ± 0.21
Kin8nm	8192	8	0.10± 0.00	0.10 ± 0.00	0.09 ± 0.00	0.09 ± 0.01	-0.90± 0.01	-0.95 ± 0.03	-1.20 ± 0.02	-1.14 ± 0.01
Naval Propulsion	11,934	16	0.01± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-3.73± 0.01	-3.80 ± 0.05	-5.63 ± 0.05	-2.65 ± 0.00
Power plant	9568	4	4.12± 0.03	4.02 ± 0.18	4.11 ± 0.17	4.20 ± 0.21	2.84± 0.01	2.80 ± 0.05	2.79 ± 0.04	2.71 ± 0.17
Protein structure	45,730	9	4.73± 0.01	4.36 ± 0.04	4.71 ± 0.06	4.34 ± 0.00	2.97± 0.00	2.89 ± 0.01	2.83 ± 0.02	2.56 ± 0.00
Wine quality	1599	11	0.64± 0.01	0.62 ± 0.04	0.64 ± 0.04	0.52 ± 0.05	0.97± 0.01	0.93 ± 0.06	0.94 ± 0.12	0.29 ± 0.05
Yacht	308	6	1.02± 0.05	1.11 ± 0.38	1.58 ± 0.48	0.82 ± 0.43	1.63± 0.02	1.55 ± 0.12	1.18 ± 0.21	0.64 ± 0.34
YearPredi MSD	515,345	90	8.88± NA	8.85 ± NA	8.89 ± NA	8.90 ± NA	3.60± NA	3.59 ± NA	3.35 ± NA	2.48 ± NA

are given linear and exponential activation for the base and uncertainty networks, respectively. We trained an ensemble of five NNs to minimize the MSE, and finetuned using another ensemble of five NNs, trained to minimize the NLL as given in equation (1), for a better learning of uncertainty, without compromising on the quality of predictions. These are implemented in Tensorflow, using default weight initializations. We used a batch size of 32 and Adam optimizer with a default learning rate.

In order to have a less biased estimate of the model skill and to prevent data leakage, we used k-fold cross validation (CV), where the dataset is split into 20 folds, after the dataset is shuffled randomly. The datasets are normalized so that the input features and the targets have zero mean and unit variance in the training set. Also to be noted that, only input features are normalized. The results of 20fold CV are summarized with the mean of the model scores obtained for prediction and uncertainty estimates. We did the train-test splitting approach only for year prediction and 5-fold CV on protein structure since they are large datasets.

Here we compare the quality of predictive uncertainty estimation as well as predictive accuracy on supervised regression tasks only. We analyzed the results using a proper loss criterion like NLL, since NLL is the popular metric for reflecting both the accuracy and quality of predictive uncertainty [31]. In table 1, we provide the results for test accuracy, NLL error on real-world datasets along with all the baselines' results reported in their respective papers. The best results are shown in bold for each dataset. We demonstrate that our framework is superior to obtaining predictive accuracy and uncertainty estimation. We can observe from the results that our method can outperform or equally compete with other baselines for all the datasets.

5. CONCLUSIONS

In this paper, a simple and extensible method to estimate the uncertainty of the output of a network by exploiting multi-task learning framework is proposed. This is achieved by exploiting the learned features from the base task to another network that captures the corresponding uncertainties. Training such a network by integrating the base and uncertainty subnetworks using a well-defined loss criterion will enable it to capture y along with its ambiguities for a given x. Intuitively, by using ensemble learning. the proposed method captures both the aleatoric (data) and epistemic (model uncertainty by averaging predictions over multiple models) uncertainties. Hence, such an approach works as a supporting tool that can augment any pretrained network with uncertainty estimation capability, without impacting their predictive performance. The superior prediction capability and well calibrated uncertainties of the proposed approach has been demonstrated on a series of benchmark experiments for supervised learning task. We hope this algorithm has a significant potential and commercial impact towards practical and accurate deep learning methods for quantifying predictive uncertainty and paves the way for future research in this direction.

REFERENCES

[1] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. of Intl. Conf. Machine Learning (ICML),* vol. 48, Jun. 2016, pp. 1050–1059.

[2] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics: Science and Systems Conf.,* Jul. 2017.

[3] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Intl. Conf. Neural Information Processing Systems (NIPS),* 2017, p. 6405–6416.

[4] I. Osband, "Risk versus uncertainty in deep learning : Bayes , bootstrap and the dangers of dropout," 2016.

[5] J. M. Hernandez-Lobato and R. P. Adams, "Probabilistic backpropagation for´ scalable learning of bayesian neural networks," in *Proc. Intl. Conf. Machine Learning (ICML),* 2015, p. 1861–1869.

[6] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proc. IEEE Intl. Conf. Neural Networks (ICNN),* vol. 1, 1994, pp. 55–60.

[7] J. M. Bernardo and A. F. Smith, *Bayesian Theory*. John Wiley & Sons, 2009, vol. 405.

[8] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. Intl. Conf. Machine Learning (ICML),* vol. 37, Jul. 2015, pp. 1613–1622.

[9] T. Chen, E. B. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," in *Proc. Intl. Conf. Intl. Conf. Machine Learning (ICML),* 2014, p. 1683—1691.

[10] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. Intl. Conf. Neural Information Processing Systems,* Cambridge, MA, USA, 2015, p. 2575–2583.

[11] D. J. C. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Comput.,* vol. 4, no. 3, p. 448–472, May 1992.

[12] ——, "Bayesian methods for adaptive models," Ph.D. dissertation, California Institute of Technology, 1992.

[13] R. M. Neal, *Bayesian Learning for Neural Networks*. Berlin, Heidelberg: Springer-Verlag, 1996.

[14] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems,* vol. 24, 2011.

[15] C. Louizos and M. Welling, "Structured and efficient variational deep learning with matrix gaussian posteriors," 2016.

[16] L. Hasenclever, S. Webb, T. Lienart, S. Vollmer, B. Lakshminarayanan, C. Blundell, and Y. W. Teh, "Distributed bayesian learning with stochastic natural gradient expectation propagation and the posterior server," *J. Mach. Learn. Res.,* vol. 18, no. 1, p. 3744–3780, Jan. 2017.

[17] A. Korattikara Balan, V. Rathod, K. P. Murphy, and M. Welling, "Bayesian dark knowledge," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[18] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust bayesian neural networks," in *Advances in Neural Information Processing Systems (NIPS),* vol. 29, 2016.

[19] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proc. Intl. Conf. Machine Learning (ICML),* Madison, WI, USA, 2011, p. 681–688.

[20] C. E. Rasmussen and J. Quinonero Candela, "Healing the relevance vector˜ machine through augmentation," in *Proc. Intl. Conf. Machine Learning,* New York, NY, USA, 2005, p. 689–696.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, Jan. 2014.

[22] Y. Li, J. M. Hernandez-Lobato, and R. E. Turner, "Stochastic expectation´ propagation," in *Proc. Intl. Conf. Neural Information Processing Systems (NIPS),* Cambridge, MA, USA, 2015, p. 2323–2331.

[23] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for bayesian uncertainty in deep learning," in *Advances in Neural Information Processing Systems,* vol. 32, 2019.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Intl. Conf. Learning Representations (ICLR),* 2013.

[25] L. Blier and Y. Ollivier, "The description length of deep learning models," in *Advances in Neural Information Processing Systems (NIPS),* vol. 31, 2018.

[26] A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov, "Pitfalls of in-domain uncertainty estimation and ensembling in deep learning," in *Intl. Conf. Learning Representations (ICLR),* 2020.

[27] S. Fort, H. Hu, and B. Lakshminarayanan, "Deep ensembles: A loss landscape perspective," 2020.

[28] M. Valdenegro-Toro, "Deep sub-ensembles for fast uncertainty estimation in image classification," 2019.

[29] W. H. Beluch, T. Genewein, A. Nurnberger, and J. M. Kohler, "The power of ensembles for active learning in image classification," in *IEEE Conf. Computer Vision Pattern Recognition (CVPR)*, 2018, pp. 9368–9377.

[30] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, "Why m heads are better than one: Training a diverse ensemble of deep networks," 2015.

[31] J. Quinonero-Candela, C. E. Rasmussen, F. Sinz, O. Bousquet, and˜ B. Scholkopf, "Evaluating predictive uncertainty challenge," in¨ *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, Berlin, Heidelberg, 2006, pp. 1–27.

[32] J. Moberg, L. Svensson, J. Pinto, and H. Wymeersch, "Bayesian linear regression on deep representations," *ArXiv*, vol. abs/1912.06760, 2019.