Project proposal

Yuchen Zhuang[1], Wenqi Shi[1], Tianyu Zhan[1], Jincheng Zhu[1], Yuechen Wu[1]
[1]Georgia Institute of Technology

Project summary

Nowadays, machine learning has earned its popularity in multiple areas with its magical performance. However, before applying machine learning techniques to circumstances where reliability is the priority, we still need to quantify their uncertainty to reduce the risk. Thanks to recent progress in Bayesian deep learning, such quantification has been made possible, which significantly contributes to improving the decision making process. For our project, we will look into the mathematical deductions of quantifying uncertainty in Bayesian models, and conduct several experiments in different application scenarios.

With the rapid development of machine learning, highly-advanced algorithms have gradually moved towards black boxes, which leads to a new problem, explain-ability. Although the highly-advanced models and algorithms are capable of learning powerful representations of mappings between high dimensional given input and expected output, at occasions even their designers cannot explain why they arrived at some mappings. In addition, current measurements could hardly explain the results from the perspective of confidence. A well-known example is the accident caused by the auto-pilot system of Tesla, where it mistook the white side of a trailer as bright sky after a difficult decision. Considering the fact that machine learning models can make mistakes, we still need to know the uncertainty of mappings to support our decision making. In other words, rather than training a dog-cat classifier with 100% validation accuracy, it is more important to train it to say "I don't know" when given a picture of fish.

We will address the problem of uncertainty in two aspects - model uncertainty and data uncertainty. Bayesian neural networks (BNN) is one of the approaches to quantify uncertainty associated with models. Data uncertainty inherits the noise captured by observation and measurement uncertainty. With sufficient analysis of uncertainties, decision making problems mentioned above will be better handled. In the project, we will also explore the quantification of both uncertainties in several scenarios, including simple data regression, natural language processing (NLP) and computer vision (CV). Besides, because the uncertainty scores we obtain is an absolute value, only comparison can make the method make sense, we also need to seek for a metric that is suitable for this situation.

1 Detailed project description

1.1 Problem description

As machine learning is developing fast nowadays, it has been utilized in many applications in different areas and has successfully aroused an increasingly number of people's attention no matter from the industrial side or from the academic side. However, we can know from a series of literature and studies that machine learning can make mistakes [1], which might enable it to cause some serious problems in areas like auto-navigation, medical diagnosis, aeronautics and so on. Thus, we need to seek for some uncertainty quantification methods to make the model output some certain metrics besides the performance of accuracy, which can tell us how uncertain the model is towards this decision.

For this problem, we mainly focus on the dropout variation Inference methods in Bayesian Neural Network. We aim to utilize this method to show and give the score of uncertainties. Hopefully, we also want to utilize this method in different applications like in regression tasks and classification tasks in Computer Vision (CV) and Natural Language Processing (NLP). And for these different areas, we also aim to give a specialized metric to judge the uncertainty scores are reasonable or not.

1.2 Related works

As uncertainty quantification has taken more and more researchers' focuses, we can see a lot of methods regarding to this task. One of the hottest solutions are the Bayesian methods, like Bayes-by-backprop [2], Monte Carlo Dropout [3], Stochastic Gradient Langevin Dynamics [4]. They are all based on the Bayesian Deep Learning and Bayesian Neural Network and the only difference is that they utilize different methods to introduce the Bayesian Inference. There are also plenty of non-Bayesian methods, like temperature scaling [5], ensemble method [6] and adversarial training methods.

Here, we choose the Bayesian methods because we think the mathematical foundations and the deducing process is suitable for this course, statistical machine learning. From these many Bayesian methods, we choose the MC Dropout for the reason that it does not need us to have great modification towards very complicated model structures, which is also the advantage of this method. However, it also has drawbacks that it might be influenced by the selection of hyper-parameters largely and it may influence the performance to some extent.

1.3 Necessary background

Many factors in machine learning can affect the uncertainties, when we apply the different models. In this paper we separate the uncertainties into 2 categories, epistemic uncertainty and aleatoric uncertainty [7], and they can also be named as model uncertainty and data uncertainty.

Before introducing more insights on these two concepts, we can define them intuitively with the law of total variance. Given an input variable x and its corresponding output variable y, the variance (total uncertainty) of y can be decomposed as:

$$\mathrm{Var}(y) = \mathrm{Var_y}(E_x[y|x]) + E_x[\mathrm{Var_y}(y|x)] \tag{1}$$

With it can be further decomposed into two components, we mathematically define model uncertainty and data uncertainty as:

$$\begin{aligned} U_m(y|x) &= \mathrm{Var_x}(E_y[y|x]) \\ U_d(y|x) &= E_x[\mathrm{Var_y}(y|x)] \end{aligned} \tag{2}$$

where U_m and U_d are model and data uncertainties respectively. We can see that both uncertainties partially explain the variance in the observation. In particular, model uncertainty explains the part related to the mapping process $E_y[y|x]$ and data uncertainty describes the variance inherent to the conditional distribution $\mathrm{Var_y}(y|x)$. By quantifying both uncertainties, we essentially are trying to explain different parts of the observation noise in y. We may explain and deduce the equations in details for several scenarios in our project.

1.3.1 Model Uncertainty (Epistemic Uncertainty)

First, we are uncertain about whether the structure choice or the selected model parameters can best describe the data distribution. This is the so-called model uncertainty, also known as epistemic uncertainty. Such kind of uncertainty accounts for uncertainty in the model parameters. It captures our ignorance about which model generated our collected data and can be explained away given enough data. Bayesian neural networks (BNN) [8–12] is one of the approaches to quantify uncertainty associated with model parameters.

Modern neural networks are parameterized by a set of model weights $\mathbf{W}$. In the supervised setting, for a dataset $D = \{(\mathbf{X}_i, y_i)\}_{i=1}^{N}$, a point estimation for $\mathbf{W}$ is obtained by maximizing certain objective function. The core part of Bayesian neural network can be interpreted as following equation:

$$P(\mathbf{W}|D) = \frac{P(D|\mathbf{W})P(\mathbf{W})}{P(D)} \tag{3}$$

BNN introduces model uncertainty by using a posterior distribution $P(\mathbf{W}|D)$ instead of a single point estimate result to describes possible values for the model weights given the dataset.

However, exact inference for BNNs is rarely available given the complicated nonlinear structures and high dimension of model parameters $\mathbf{W}$ in modern neural networks. Also the distribution of data $P(D)$ represents the real data distribution, which is theoretically impossible to obtain. Hence, various methods are proposed to approximate the inference [2, 3, 13–15]. Among them, the Monte Carlo Dropout (MC Dropout) requires minimum modification to the original proposed model and is easy to realize. Dropouts are applied between non-linearity layers in the network and are activated at test time which is different from a regular dropout. They showed that the process is equivalent to variational Bayesian approximation where the approximation distribution is a mixture of a zero-mean Gaussian and a Gaussian with small variances. As a result, model uncertainty can be approximately evaluated by finding the variance of the model outputs from multiple forward passes.

MC Dropout is what we will mainly use to measure model uncertainty in our project. The model uncertainty quantification metric and interpretation will be discussed in several different scenarios.

1.3.2 Data Uncertainty (Aleatoric Uncertainty)

Another case is that we may also get some noise in the collected data, which means that our training set or testing set may be some-how noisy. This often increases when we rely more on the observations and measurements are precise, noise may be generated in the data creation or transmission process. Such uncertainties are classified as data uncertainties or aleatoric uncertainty. Aleatoric uncertainty captures noise inherent in the observations. This could be for example sensor noise or motion noise, resulting in uncertainty which cannot be reduced even if more data were to be collected. Depending on whether the uncertainty is input independent, data uncertainty is further divided into homoscedastic uncertainty and heteroscedastic uncertainty. Homoscedastic uncertainty keeps the same all over the input space, which can be caused by systematic observation noise. On the contrary, the heteroscedastic uncertainty is dependent on the input.

To be more general, in our project, we will choose the heteroscedastic uncertainty and make the assumption that data uncertainty is dependent on the input. To achieve this, we need to have a model that not only predicts the output values but also estimates the output variances given some input. In other words, the model needs to give an estimation of $\mathrm{Var}_{\mathrm{y}}(y|x)$.

1.4 Specific Objectives

In this project, we plan to explore the mathematical deductions of quantifying uncertainty in Bayesian methods and its applications to some simple experiments. We separate the whole uncertainty into two categories: aleatoric (data) uncertainty and epistemic (model) uncertainty. We aim to present a Bayesian deep learning framework combining the aleatoric uncertainty and epistemic uncertainty and demonstrate the model's practicality through several tasks on different scenarios.

With the aid of the uncertainty quantification and separation, we can also measure confidence levels in some basic NLP and CV tasks like text classifications and semantic segmentation. As NLP and CV are more like using high-dimensional data we cannot just plot a simple figure showing the uncertainty quantification. We need to put forward new metrics showing this.

As NLP is one of the hottest topic in machine learning and grabs more and more people's eyes, it is very worthy quantifying the uncertainty in text analysis tasks. Sentiment analysis is a basic and popular topic of text analysis. Conventionally, sentiment analysis is done with classification. In this study, to explore the effect of quantifying uncertainties, we plan to consider both regressions and classification settings for sentiment analysis. In the regression setting, we can treat the class labels as numerical values and aim to predict the real value score given a review document. In the evaluation part of NLP task, we wish to use accuracy in the classification setting and mean square error (MSE) in the regression setting to evaluate model performances. Accuracy is a standard metric to measure classification performances and MSE measures the average deviation of the predicted scores from the true ratings and is defined as:

$$\text{MSE} = \frac{\sum_{i=1}^{N} (\text{gold}_i - \text{predicted}_i)^2}{N}. \tag{4}$$

Another metric is to see the model's calibration, which is the Expected Calibration Error (ECE):

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\operatorname{acc}(B_m) - \operatorname{conf}(B_m)|. \tag{5}$$

where the $\operatorname{acc}(B_m)$ and $\operatorname{conf}(B_m)$ can be defined as:

$$\begin{aligned} \operatorname{acc}(B_m) &= \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i), \\ \operatorname{conf}(B_m) &= \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i. \end{aligned} \tag{6}$$

Quantifying uncertainty in computer vision applications can be largely divided into two categories: regression settings and classification settings [16]. In our application to CV, we plan to evaluate our model with pixel-wise depth regression in regression settings and semantic segmentation in classification settings. More specifically, we can model the benefit of combining both epistemic uncertainty as well as aleatoric uncertainty to improve the performance and show the robustness of our proposed method. Hopefully, we wish to study the effectiveness of modeling aleatoric and epistemic uncertainty and give a detailed analysis of the performance with single uncertainty and combination uncertainty model as well. In particular, we wish to explore what they capture via observing and quantifying the visualization results and performance of these uncertainty measurements.

2 List of tasks/collaboration plan

2.1 List of tasks

We can summarize from the specific objectives for this part. The list of tasks can be illustrated as below:

1. For the mathematical foundations, we want to have a brief basic view about the Bayesian Neural Network and get familiar with the Monte Carlo Dropout, the specific method for uncertainty quantification.

2. With the mathematical basis, we can utilize this method in many applications of deep learning to see the performance and make comparison in different areas to see some reasonable impacts.

3. For regression task, we aim to establish a code on the toy sin dataset to see the quantification of the different kinds of uncertainties in the given method.

4. For classification task, we choose the hottest two directions, Natural Language Processing and Computer Vision to seek for some experiments and applications.

2.2 Tasks assignment and progress timetable

The whole timetable of the progress is defined as below:

1. From **March 6th - March 20th**, we need to learn the mathematical foundations, read related works and have group discussions.
2. From **March 21st - April 7th**, we need to separate the different applications tasks to different members to realize and we can cooperate and have discussions in this procedure.
3. From **April 7th - April 16th**, we need to make modifications to our accomplishment and make the poster for presentation.
4. From **April 17th - April 24th**, we need to write the final report of this project and we may also make some correction or improvements according to Prof. Bloch and the other classmates' ideas and opinions.

And the work assignments and leaders (mainly responsible) for different parts can be briefly set as below:

1. For the mathematical foundation deduction and literature study: **All the members**;
2. For the regression task accomplishment: **Jincheng Zhu** & **Yuechen Wu**;
3. For the Natural Language Processing classification task: **Yuchen Zhuang**;
4. For the Computer Vision applications: **Wenqi Shi** & **Tianyu Zhan**;
5. Literature writing (poster and report): **All the members**;
6. Check and summary: **Yuchen Zhuang**;
7. Presentation: **TBD**
8. Although the assignments have leaders or people that are mainly responsible for the tasks, we still have flexibility in these sections and these are just brief assignments.

2.3 Potential challenges

(a) The model's uncertainty is not calibrated. A calibrated model means the predictive probabilities match the empirical frequency of the data. Through the derivation's relation to Gaussian processes [3], we can see the lack of calibration. Gaussian processes' uncertainty is known to not be calibrated-the Gaussian process's uncertainty depends on the covariance function chosen, which is shown in [3] to be equivalent to the non-linearities and prior over the weights. The choice of a Gaussian process's covariance function follows from our assumptions about the data. For example, when the model's uncertainty should increase far from the data, we might choose the squared exponential covariance function.

(b) This method requires that there are dropout layers in the network architecture. If there are not dropout layers in it, we cannot use this method. To overcome that, we can add a dropout layer to the initial network when do the testing or choose the existing net work architecture which has the dropout layer.

(c) The training time of our model is identical to that of existing models in the field, but the test time is scaled by the number of averaged forward passes through the network. To overcome that, distributed hardware can transfer an input to a GPU and set a mini-batch composed of the same input multiple times which allows us to obtain MC estimates in constant time almost trivially.

(d) Uncertainty is a novel field in deep learning and has not been widely applied at present. The related literature is not abundant. To overcome that, we will read the existing classic literature carefully and try to understand them deeply. Then apply it in different network architecture in different field like NLP, CV to see the effect.

References

[1] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[2] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[3] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[4] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 681–688.

[5] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1321–1330.

[6] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[7] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? does it matter?" *Structural safety*, vol. 31, no. 2, pp. 105–112, 2009.

[8] W. L. Buntine and A. S. Weigend, "Bayesian back-propagation," *Complex systems*, vol. 5, no. 6, pp. 603–643, 1991.

[9] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.

[10] ——, "Probable networks and plausible predictions—a review of practical bayesian methods for supervised neural networks," *Network: computation in neural systems*, vol. 6, no. 3, pp. 469–505, 1995.

[11] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[12] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[13] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*, 2011, pp. 2348–2356.

[14] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *International Conference on Machine Learning*, 2015, pp. 1861–1869.

[15] L. Zhu and N. Laptev, "Deep and confident prediction for time series at uber," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 103–110.

[16] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NIPS*, 2017.