Uncertainty Quantification in deep learning: Literature Review and Taxonomy

Hanyu Li

2020 年 4 月 20 日

1 epistemic uncertainty

1.1 definition

Epistemic uncertainty accounts for uncertainty in the model parameters –uncertainty which captures our ignorance about which model generated our collected data. [1]e.g. High when a model faces an image that isn't in any class(will mitigate given enough data)

1.2 how to derive

1.2.1 MC sampling methods

The MC samples are generally computed using an ensemble of neural networks. The prediction ensemble could either be generated by differently trained networks, or by keeping drop-out at test-time[2].

an example: MC dropout Yarin Gal et al. showed that a neural network with arbitrary depth and non-linearities, with dropout applied before every weight layer, is mathematically equivalent to an approximation to a well known Bayesian model(deep Gaussian process model)[3]. Thus they can derive the uncertainty simply by collecting the results(mean for predictive mean and variance for uncertainty) of stochastic forward passes through the model.

pros and cons It is a practical approach for approximate inference in large and complex models because it does not requires more parameters and sometimes not even more time complexity. However, the dropout approximation fails in some network architectures[4]. Also, the prior are often fixed. For example, MC dropout can only works when the prior satisfies the assumption of bernoulli distribution.

1.2.2 variational methods

Bayesian Neural Network(BNN) automatically yield us the prediction uncertainties because it assumes that its weight parameters w are random variables under certain distributions. Thus after we achieve the posterior $p(w|X, Y)$, where X, Y are the given dataset, the uncertainty can be derived from the prediction y which is based on a given x. Here y is also a random variable. The variational inference way of training BNN is the process of pushing these distributions $q_\theta(w)$ to be as close as possible to the posterior $p(w|X, Y)$ and θ is the parameters of the distribution of w, which will be constant after training process. Through this process, it converts the inference problem into an optimization problem.

an example: Bayes by Backprop In one of the practical algorithms, Bayes by Backprop[5], they use back propagation just like normal NNs, but it is the parameters of $q_\theta(w)$, θ, that are updated using reparamaterization trick and the loss derived from the objective function ELBO. ELBO is ”evidence lower bound”. By maximizing this we can get the $q_\theta(w)$ that is close to $p(w|X, Y)$.

pros and cons It optimises a well-defined objective function to learn a distribution on the weights of a neural network. However in practice, Bayesian NNs are often harder to implement and computationally slower to train compared to non-Bayesian NNs[6].

2 aleatoric uncertainty

2.1 definition

Aleatoric uncertainty captures noise inherent in the observations.[1] E.g. High when data has noise that cannot be explained by the real correlation of X and Y(will not mitigate given data because new data also has this kind of noise) We can use a single network to transform the input x, with its head split to predict both output y as well as aleatoric uncertainty. We can update the parameters with a specific loss to achieve intended function[1].

2.2 how to derive

Kendall et al. proposed a way to learn how to yield aleatoric(data) uncertainty from the data itself. They feed the input $\mathbf{x}$ into one network, with its head split to predict both $\hat{\mathbf{y}}_i$ as well as the data uncertainty σ_i. The loss will be as below(D is the number of pixels in image segmentation task in the paper):

$$\mathcal{L}_{BNN}(\theta) = \frac{1}{D}\sum_i \frac{1}{2}\hat{\sigma}_i^{-2}\left\|\mathbf{y}_i - \hat{\mathbf{y}}_i\right\|^2 + \frac{1}{2}\log\hat{\sigma}_i^2$$

Here the parameters in the σ_i end is not a random variable because we do not need uncertainty for uncertainty. The loss function, though derived from mathematical deduction, can be explained straightfowardly.

参考文献

[1] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *CoRR*, abs/1703.04977, 2017.

[2] Mattia Segù, Antonio Loquercio, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *CoRR*, abs/1907.06890, 2019.

[3] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *arXiv preprint arXiv:1506.02142*, 2015.

[4] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference, 2015.

[5] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks, 2015.

[6] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017.