Quantifying Uncertainty with Probabilistic Machine Learning Modeling in Wireless Sensing

Amit Kachroo
Amazon Lab126
Sunnyvale, California, USA
amkachro@amazon.com

Sai Prashanth Chinnapalli
Amazon Lab126
Sunnyvale, California, USA
saic@amazon.com

Abstract—The application of machine learning (ML) techniques in wireless communication domain has seen a tremendous growth over the years especially in the wireless sensing domain. However, the questions surrounding the ML model’s inference reliability, and uncertainty associated with its predictions are never answered or communicated properly. This itself raises a lot of questions on the transparency of these ML systems. Developing ML systems with probabilistic modeling can solve this problem easily, where one can quantify uncertainty whether it is arising from the data (irreducible error or aleatoric uncertainty) or from the model itself (reducible or epistemic uncertainty). This paper describes the idea behind these types of uncertainty quantification in detail and uses a real example of WiFi channel state information (CSI) based sensing for motion/no-motion cases to demonstrate the uncertainty modeling. This work will serve as a template to model uncertainty in predictions not only for WiFi sensing but for most wireless sensing applications ranging from WiFi to millimeter wave radar based sensing that utilizes AI/ML models.

Index Terms—probabilistic modeling, Bayesian networks, wireless sensing, WiFi, uncertainty quantification, machine learning

I. INTRODUCTION

The application of artificial intelligence (AI)/machine learning (ML) algorithms in wireless sensing applications is seeing an immense growth over the years, and in-fact these models are now embedded into real-world products and features. The main advantages of utilizing AI/ML techniques over conventional principle techniques is the reduced computation complexity, increased energy efficiency, and better optimal solutions. However, one of the biggest challenges associated with these AI/ML models is in its inference reliability or put simply, how confident is the model in its predictions. This unfortunately has not been clearly understood or quantified in the domain of AI/ML in wireless sensing area. In this work, we will discuss the method to include uncertainty (aleatoric and epistemic) into the ML model with an example of WiFi sensing [1].

II. UNCERTAINTY IN AI/ML MODELS

To start with, the term uncertainty actually in itself means lack of knowledge to a particular outcome. In AI/ML domain, this can be attributed to either data/features itself (measurement noise, or wrong labeling) or to the model (model parameters) or lack of training data. This is broadly classified as aleatoric and epistemic uncertainty [2].

- **Aleatoric or indirect uncertainty**- This type of uncertainty arises from the unaccounted factors, such as environment settings, noise in the input data, or bad input feature selections. It is also known as an irreducible error and can’t be remediated with more data.
- **Epistemic or direct uncertainty**- This type of uncertainty usually arises from the lack of knowledge about the model or data. One example can be over-generalization, where the ML model is very complex as compared to the amount of data it is trained on and thereby overgeneralizes on a test dataset.

III. METHODS TO MODEL UNCERTAINTY

The methods to integrate aleatoric and epistemic uncertainty are described in Algorithm-1 and Algorithm-2 as,

**Algorithm 1** method to measure Aleatoric uncertainty

**Input:** \( D(X_i, y_i) \), output layer as probabilistic node: \( N(\mu, \sigma) \), optimizer: rmsProp

**Output:** \( \hat{y}_i \)

1: for \( epoch = 1 \) to \( num\_epochs \) do
2: i) Calculate loss and gradients using negative log-likelihood (NLL)
3: ii) Apply gradients, update weights and monitor loss
4: end for
5: Determine \( \mu \) and \( \sigma \) from the output layer

**Algorithm 2** method to measure Epistemic uncertainty

**Input:** \( D(X_i, y_i) \), optimizer: rmsProp

**Output:** \( \hat{y}_i \)

**Initialization**

a) Prior distribution: \( p(\theta) \) to weights \( \theta \). b) Variational posterior: \( q(\theta|\phi) \) with trainable parameter \( \phi \).

2: for \( epoch = 1 \) to \( num\_epochs \) do
3: i) Calculate loss and gradients using NLL
4: ii) Learn \( \phi \) using Kullback-Leibler divergence
5: iii) Apply gradients, update weights and monitor loss
6: end for
7: Determine class probabilities or mean/variance.
To combine both aleotoric and epistemic uncertainty in a AI/ML model, we can integrate these probabilistic layers with weights as random variable and a probabilistic output layer.

IV. ML MODELING AND RESULTS

For our experiments, we will create a very simple two probabilistic model layers with the first layer having 4 nodes, and second one with 2 nodes. The output layer is modeled as a probabilistic layer to capture the aleotoric uncertainty. The model is then trained on 3-homes for 200 epochs while testing on one left out home. Figure 1 shows one example from two test homes-1 and -3, while being trained on the rest 3 homes. As one can observe, the model trained on homes-2, 3, and 4, is very certain on his prediction in that one example data from home-1, while the model trained on homes-1,2, and 4, is very uncertain in his prediction in that one example data from home-3. To analyze the model’s uncertainty across the full test set, we can use entropy of the distribution as a metric for uncertainty. The entropy is given as, 

\[ H_i = -\sum_{j=1}^{n_i} p_i \log_2(p_i) \]

where \( n_i \) is the number of samples of a class and \( i \) represent one of those samples. Thus, higher the entropy of a class, higher is the uncertainty.

Table I presents the mean entropy for each class in all the test home cases. There are few important observation from these results,

- Although the accuracy for test home-1 is \( \sim 80\% \) but the mean entropy for motion class is very high as compared to no-motion class. This implies that the model is very uncertain in its prediction on motion classes as compared to no-motion class. Same can be inferred for test home-4.
- Test home-2 and home-3 are having equal mean entropies across classes but the accuracy is lower in home-3, which means the model is performing really bad in terms of predictions for both the classes and is very uncertain in its predictions as compared to other homes especially for no-motion class.
- The best performance of the model is on test home-4 in the no-motion class.

V. CONCLUSION AND FUTURE WORK

In this paper, we describe the need of uncertainty quantification of AI/ML models and laid the foundations for modeling the two types of uncertainty (aleotoric and epistemic uncertainty) in a AI/ML model. The results highlight the need of uncertainty quantification, where we observed that even though in test Home-1, the prediction accuracy was high but the predictions for motion class as compared to no-motion class were very uncertain. We also saw the case of test Home-3 having high epistemic as well as high aleotoric uncertainty with less prediction accuracy. These results are valuable as it forces one to rethink on the data collection strategy or feature engineering to handle aleatoric uncertainty or the need to collect more data or iterate over different model architecture to overcome epistemic uncertainty.

REFERENCES

[1] A. Kachroo, J. Park, and H. Kim, “Channel assignment with transmission power optimization method for high throughput in multi-access point WLAN,” in 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), 2015, pp. 314–319.
[2] Y. Gal, “Uncertainty in deep learning,” Ph.D. dissertation, University of Cambridge, 2016.