Uncertainty Estimation with Bayesian Neural Networks for Parkinson's Disease Diagnosis

Adi Pasic, Maxwell Johnson, Ari Economon, and Juan Leyva
Department of Electrical and Computer Engineering
The University of Houston, Houston, Texas

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Estimating uncertainty in machine learning is an essential feature for building robust, interpretable systems in error-sensitive contexts such as health and medicine. These applications utilize deep learning architectures for prognosis and classification of conditions ranging from diabetic retinopathy to cancerous lesions. In a medical context, interpretability means that the diagnosis can be rationalized by identifying uncertainty in the prediction and isolating variables causing the uncertainty. In effect, interpretability allows the model to be treated as a black-box, facilitating the incorporation of such systems in contexts where the operator is not necessarily experienced with machine learning.

A. Bayesian Networks

A major research focus for Bayesian Neural Networks (BNNs) is identifying features that propagate uncertainty [4] [6]. Since BNNs take a probabilistic approach to the process of determining network parameters that minimize some objective function. Instead of representing weights with point estimates, a probability distribution is found for each weight that the network then samples from during a forward pass. The Bayesian approach is to update this posterior distribution with every observation. Given a dataset D and input x, the network seeks to learn an optimal distribution for $p(y|x, D)$, where y is the true class. This distribution is represented in closed form as

$$p(y|x, D) = \int p(y|x, \theta')p(\theta'|D)\, d\theta' \tag{1}$$

However, the posterior $p(\theta'|D)$ is intractable to find, so approximation techniques such as Markov Chain Monte Carlo and variational inference are used to approximate the true posterior or learn parameters of an intermediate distribution. However, these inference techniques do not scale to deep neural network architectures [2].

B. Monte Carlo Dropout

Monte Carlo Dropout (MC Dropout) extends the classical dropout regularization technique to Bayesian inference by applying dropout at test time, resulting in a distribution for the network prediction. It scales well to large deep learning architectures and can be used to approximate Bayesian inference in otherwise deterministic systems. The ensemble of all dropout networks aggregates to the posterior generated by variational inference, allowing for quick and native sampling of network parameters [3]. MC Dropout has been criticized for only partially capturing uncertainty of model predictions but provides an efficient approximation of variational inference for large networks [2].

II. Methodology

We utilize a dataset of preprocessed vocal data from a balanced number of patients with and without Parkinson's Disease [1]. There are a total of 1040 datapoints, composed of multiple recordings of 40 patients: 6 female, 14 male patients diagnosed with Parkinson's Disease, and 10 female, 10 male healthy patients. The various sounds included sustained 3 sustained vowels, numbers 1-10, 4 short sentences, and 9 words. From each sound recording, 26 linear time frequency based features were obtained: Jitter (local), Jitter (local, absolute), Jitter (rap), Jitter (ppq5), Jitter (ddp), Shimmer (local), Shimmer (local, dB), Shimmer (apq3), Shimmer (apq5), Shimmer (apq11), Shimmer (dda), AC, NTH, HTN, Median pitch, Mean pitch, Standard deviation, Minimum pitch, Maximum pitch, Number of pulses, Number of periods, Mean period, Standard deviation of period, Fraction of locally unvoiced frames, Number of voice breaks, Degree of voice breaks, and Unified Parkinson's Disease Rating Scale (UPDRS). For feature acronym attribute explanations, see Appendix A.

To remove bias in the training dataset, we randomly generate a 90-10 train-test split. The data are also scaled using mean normalization to improve network loss.

A. Estimating Uncertainty

We take the predictive difference approach in estimating the network's sensitivity to individual features. This method compares network output probabilities after eliminating a single feature x_i from the test input x, represented as $(x \setminus x_i)$. The difference in probability

$$\text{probDiff}_i(y|x) = p(y|x) - p(y|x \setminus x_i) \tag{2}$$

is used to estimate the change in predictive uncertainty of the output [6]. There are multiple methods of eliminating features, which are highly dependent on the type of network being used. A naïve Bayesian classifier natively supports unknown values, but other networks require a special unknown value to take the place of the eliminated feature. Feature elimination can be achieved by approximating a marginalization over feature x_i [3] [6]. Given all m_i possible values for x_i, feature elimination can be expressed as

$$p(y|x \setminus x_i) = \sum_{s=i}^{m_i} p(y|x \leftarrow x_i = x_s) p(x_i = x_s | x \setminus x_i) \quad (3)$$

The term $p(y|x \leftarrow x_i = x_s)$ represents the output probability when the i^{th} term of x is replaced by the sample x_s. The distribution $p(x_i|x \setminus x_i)$ can be approximated by $p(x_i|x)$. A further approximation of 3 from [3] uses Monte Carlo sampling to approximate the change in predictive uncertainty.

$$\Delta U_{i,predictive}(x) = \mathbb{H}[y|x, D] - \mathbb{H}[y|x \setminus x_i, D] \quad (4)$$

$\mathbb{H}[y|x, D]$ is evaluated by MC sampling from the approximation of the posterior distribution given by MC dropout:

$$\mathbb{H}[y|x, D] \approx -\sum_c \left(\frac{1}{K} \sum_k p(y = c|x) \right) \log \left(\frac{1}{K} \sum_k p(y = c|x) \right) \quad (5)$$

Where c is the number of classes. When x_i is unknown, $\mathbb{H}[y|x \setminus x_i, D]$ is approximated by

$$\mathbb{H}[y|x \setminus x_i, D] \approx -\sum_c \left[\frac{1}{MK} \sum_{m=1}^{M} \sum_{k=1}^{K} p(y = c|x \setminus x_i, x_i^m) \right] \times \log \left[\frac{1}{MK} \sum_{m=1}^{M} \sum_{k=1}^{K} p(y = c|x \setminus x_i, x_i^m) \right] \quad (6)$$

In addition to performing K forward passes, M samples from the distribution $p(x_i)$ are taken and substituted for the i^{th} term of x, with a sample denoted as x_i^m.

The uncertainties are decomposed into aleatoric and epistemic uncertainties by the technique introduced in [3], where Monte Carlo sampling is done on the network parameters to get the expected value of the output entropy. Aleatoric uncertainty is a measure of how much the feature value contributes to output uncertainty, while epistemic uncertainty indicates that the model parameters are the cause of uncertainty. While epistemic uncertainty can be alleviated by more data, aleatoric uncertainty indicates the absence of essential features that the model needs to make a confident decision [3].

$$\Delta U_{i,aleatoric} = E\left(\mathbb{H}[y|x]\right) - E\left(\mathbb{H}[y|x \setminus x_i]\right) \quad (7)$$

When x_i is known, K samples are averaged

$$E\left(\mathbb{H}[\hat{y}|x, w]\right) \approx -\frac{1}{K} \sum_k \left[\sum_c p(y = c|x,) \log p(y = c|x,) \right] \quad (8)$$

When x_i is unknown, M samples are taken from its distribution

$$E\left(H[y = c|x \setminus x_i, w]\right) \approx -\frac{1}{K} \sum_k \left[\sum_c \left(\frac{1}{M} \sum_m p(y = c|x_i^m, x \setminus x_i,) \right) \times \log \left(\frac{1}{M} \sum_m p(y = c|x_i^m, x \setminus x_i,) \right) \right. \quad (9)$$

The epistemic uncertainty is the difference between predictive and aleatoric uncertainties, but [3] does include a derivation.

We train on a deep neural network of three hidden layers with 20, 16, and 4 units, respectively. These layer sizes were chosen as they had the greatest test accuracy of those tested. The dropout rate is varied between 0.1, 0.3, and 0.5. Predictive, aleatoric, and epistemic uncertainties are computed for each feature, and distributions for incorrect predictions are compared with correct predictions. The following analysis is done on a network with a dropout rate of 0.1 since it yielded the lowest training loss.

III. Results

A network of 3 hidden dropout layers, with 20, 16, and 4 units respectively, was trained for 100 epochs at dropout rates of 0.1, 0.3, and 0.5. The training loss for each is shown in 1. The remainder of the results are obtained from a network with a dropout rate of 0.5 since it yielded the highest test accuracy and would be the favorable choice in practice.

Predictive uncertainties for each feature are displayed in Figure 2, along with its decomposition into aleatoric and epistemic uncertainties. We observe that every feature has a significant epistemic uncertainty, with a complementary aleatoric uncertainty that results in a total predictive uncertainty of close to 0.

Given this observation, we plot the distributions of our model's correct and incorrect predictions to infer its weaknesses and expose patterns in classification. Shown in Figure 3 of all 26 feature distributions, segmented by whether the network was able to generate the correct prediction. We note that the distributions for jitter features are more compact for output errors, indicating that the network is interpreting deviations from the mean correctly, but is ambiguous when the inputs are close to 0.

For those features with highest epistemic uncertainty, distributions are again segmented by network correctness and compared in Figure 3. We observe that the mean period, fraction of locally unvoiced frames, and number of voice breaks have the

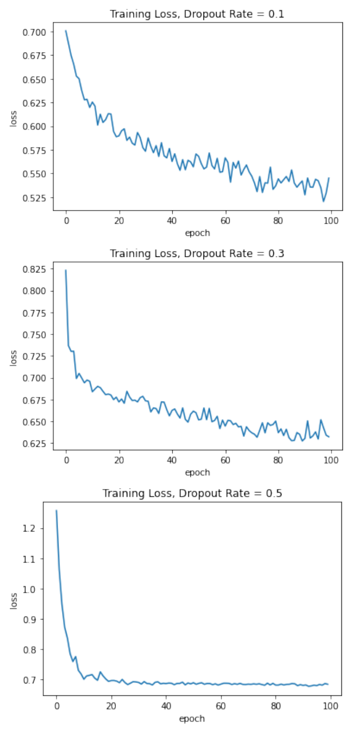

Fig. 1: Network loss during training with dropout rates of A) 0.1 B) 0.3 C) 0.5. The testing losses for each network are 56%, 58%, and 64%, respectively.

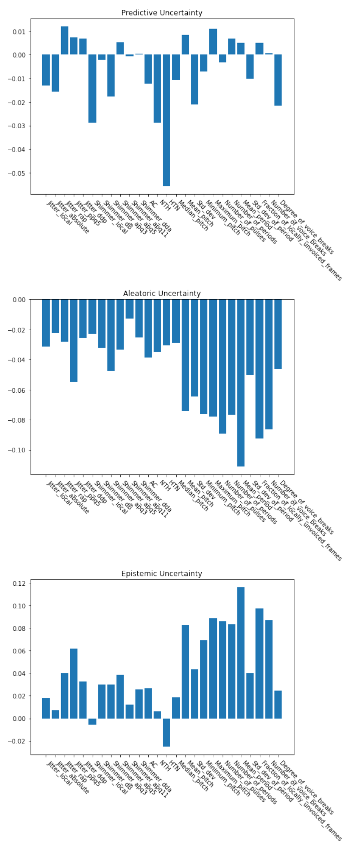

Fig. 2: Uncertainties of each feature. From top to bottom: A) Predictive, B) Aleatoric, C) Epistemic.

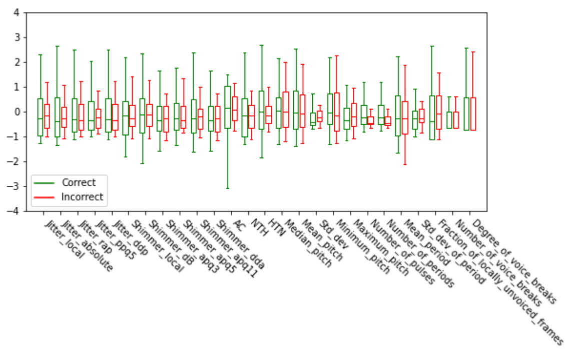

Fig. 3: Segmentation of input feature distributions by network prediction accuracy with dropout rate of 0.1.

highest epistemic uncertainties. Their distributions are shown in Figure 4. We note that the distributions for mean period and number of voice breaks are distinguishable for correct and incorrect network predictions, again indicating that the network is identifying deviations from the mean correctly, but struggles for average or near-average inputs. The compact distribution of the fraction of locally unvoiced frames indicates that the model is not generalizing well on feature values in the range -1 to 0, but again is accurate for outliers. These patterns support the high epistemic uncertainty as they indicate the model's failure to generalize on inputs confined to a specific range.

A. *Removing Correlating Features*

The network was retrained with highly correlated features ($\rho > 0.8$) removed to observe the network's behavior when we eliminate dependent features. The correlation matrix is shown in Figure 5, where we observe high correlation between the shimmer and jitter features.

Typically, removing highly correlated features brings a multitude of benefits. To list a few, removing correlated features will potentially improve the speed at which the network learns, decrease additional bias, and increase stability in the output. Due to these reasons, we thought it beneficial to see the impact on our network to see if the test accuracy of the network would increase with highly correlated features removed. Elimination of features with 80 percent correlation or higher resulted in 9 remaining features. The reduction in correlated features did not

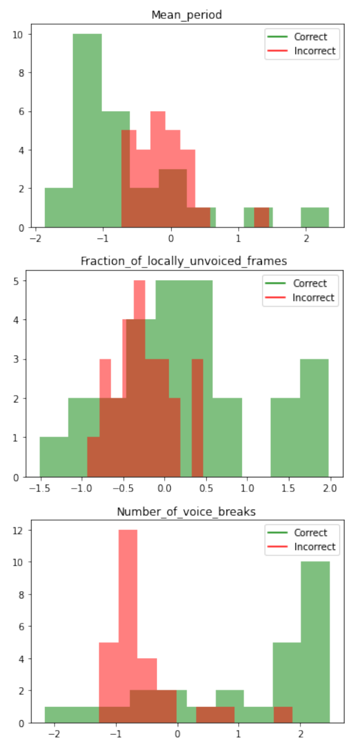

Fig. 4: Comparison of distributions for features of highest uncertainty, p=0.1.

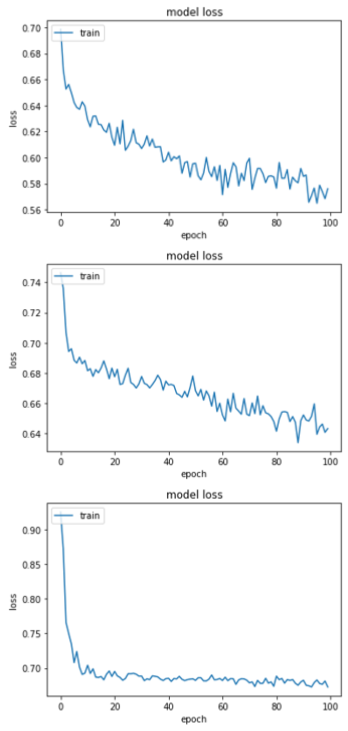

Fig. 6: Loss of model training with highly correlated features removed.

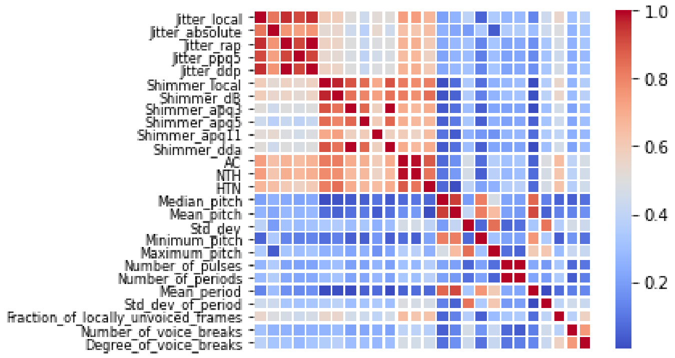

Fig. 5: Correlation matrix of all 26 features

improve the accuracy of the network and this could possibly be due to the correlated features not negatively affecting the original model. Correlated features have the ability to worsen a model, but that does not mean they will always negatively affect the model.

IV. Conclusion

Our findings for predictive uncertainty strongly indicate that the model, not the features, is responsible for uncertainty. Closer inspection of feature distributions agrees with the hypothesis that the model is not generalizing well to the testing data. Failure to generalize is supported by the high epistemic uncertainties for each feature, indicating that more testing data is required for the model to make an accurate prediction. However, the model could also be improved by instead training with native variational inference, instead of the MC-Dropout approximation. Model-specific improvements include changing the number of hidden layers, changing the number of hidden units in each layer, utilizing heterogeneous activation functions, and removing dropout from the last layer.

There are several limitations of our methodology for determining predictive uncertainty. First, we naively assume that all features are independent when computing their respective distributions, but this is not the case for the features which rely on common attributes such as shimmer and jitter. Second, our Monte Carlo sampling is limited to 20 forward passes and 20 feature samplings . Increasing these would improve consistency of results and aid in reproducibility.

References

[1] Erdogdu Sakar, B., Isenkul, M., Sakar, C.O., Sertbas, A., Gurgen, F., Delil, S., Apaydin, H., Kursun, O., ’Collection and Analysis of a Parkinson Speech Dataset with Multiple Types of Sound Recordings’, IEEE Journal of Biomedical and Health Informatics, vol. 17(4), pp. 828-834, 2013.

[2] Jospin, Laurent Valentin, et al. Hands-On Bayesian Neural Networks—a Tutorial for Deep Learning Users. arXiv preprint arXiv:2007.06823 (2020)

[3] Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In International Conference on Machine Learning, pages 1050-1059.

[4] L.R. Chai. (2018). Uncertainty Estimation in Bayesian Neural Networks and Links to Interpretability. Master’s Thesis, University of Cambridge.

[5] Depeweg, S., Hernandez-Lobato, J. M., Udluft, S., and Runkler, T. (2017). Sensitivity Analysis for Predictive Uncertainty in Bayesian Neural Networks. arXiv preprint arXiv:1712.03605

[6] Robnik-Šikonja, M. and Kononenko, I. (2008). Explaining classifications for individual instances. IEEE Transactions on Knowledge and Data Engineering, 20(5):589–600.

V. Appendix A

Descriptions for features with similar names or represented by acronyms are listed below.

1) Jitter (local): Represents the average absolute difference between two consecutive periods, divided by the average period.
2) Jitter (local, absolute): Represents the average absolute difference between two consecutive periods and is known as jitta.
3) Jitter (rap): Represents the average for the disturbance, i.e., the average absolute difference of one period and the average of the period with its two neighbors, divided by the average period.
4) Jitter (ppq5): Represents the ratio of disturbance within five periods, i.e., the average absolute difference between a period and the average containing its four nearest neighbor periods, i.e. two previous and two subsequent periods, divided by average period.
5) Jitter (ddp): Denotes the average absolute difference of differences between jitter cycles.
6) Shimmer (local): Represents the average absolute difference between the amplitudes of two consecutive periods, divided by the average amplitude.
7) Shimmer (local, dB): Represents the average absolute difference of the base 10 logarithm of the difference between two consecutive periods and it is called ShdB. Shimmer (apq3): represents the quotient of amplitude disturbance within three periods, in other words, the average absolute difference between the amplitude of a period and the mean amplitudes of its two neighbors, divided by the average amplitude.
8) Shimmer (apq5): Represents the ratio of perturbation amplitude of five periods, in other words, the average absolute difference between the amplitude of a period and the mean amplitudes of it and its four nearest neighbors, divided by the average amplitude.
9) Shimmer (apq11): Represents the ratio of perturbation amplitude of 11 periods, in other words, the average absolute difference between the amplitude of a period and the mean amplitudes of it and its 10 nearest neighbors, divided by the average amplitude.
10) Shimmer (dda): The average absolute differences between the amplitudes of consecutive periods.