Uncertainty in Neural Networks

Neural network model predictions can be underconfident (when the predicted confidence of the model is less than the true likelihood of the event) or overconfident, that means the model is badly calibrated. This marks the importance of having the uncertainty estimate for the neural networks.

Uncertainties can be of different types:

1. **Aleatoric uncertainty (Data-Dependent Uncertainty)**: It arises from inherent randomness or variability in the data itself, such as measurement noise or inherent variability in the data distribution. This is irreducible uncertainty as it arises from the natural complexity of the data, such as class overlap, label noise, homoscedastic and heteroscedastic noise.

2. **Epistemic uncertainty (Model-Dependent Uncertainty) :** It measures the uncertainty in estimating the model parameters given the training data - this measures how well the model is matched to the data. Thus, it reflects the model's inability to perfectly capture the underlying relationship between the input features and the target variable. Model uncertainty is reducible as the size of training data increases.

3. **Distributional Uncertainty:** It arises due to a mismatch between the training and test distributions which often happens in real life situations. This often happens when the model is unfamiliar with the test data and thus cannot confidently make predictions.

In the Bayesian framework distributional uncertainty, or uncertainty due to mismatch between the distributions of test and training data, is considered a part of model uncertainty.

In another work of Prior Networks [2], they are explicitly constructed to capture data uncertainty and distributional uncertainty. Here data uncertainty is described by the point-estimate categorical distribution μ and distributional uncertainty is described by the distribution over predictive categoricals, P(μ|x* , θ). Here x* is the new test data and θ are the model parameters.

EVALUATING DATA AND MODEL UNCERTAINTY

1. **Data Uncertainty (Aleatoric Uncertainty) :**

- To evaluate data uncertainty separately from model uncertainty, we can examine the variability of the predictions across different subsets of partitions of the data.
- We can use perturbation methods, such as adding noise or altering input features slightly, to generate multiple predictions for each data point and assess the variability of predictions.
- For a Neural Network, we can train a network where after the pre-final layer we can attach two separate neural network layers that gives us the estimate of the mean and log-variance values of the logits. The exponential of the log-variance values of the logits can give us an estimate of data uncertainty.
- We can train a Neural Network that gives an estimate of the true mean($\mu x^{\wedge}$) and variance ($\sigma_X^{\wedge}$) of the target distribution. Following loss function was used to train for the regression task [3]:

$$\mathcal{L}_{\text{NN}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma(\mathbf{x}_i)^2}||\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i)||^2 + \frac{1}{2}\log\sigma(\mathbf{x}_i)^2$$

- For classification task, we train the network with the below loss function: [4]

$$\hat{\mathbf{x}}_t = \hat{\mathbf{y}} + \boldsymbol{\varepsilon}_t \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\hat{\boldsymbol{\sigma}}^2)) \qquad \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{T}\sum_{t=1}^{T} \text{Cross entropy}(\mathbf{x}, \hat{\mathbf{x}}_t)$$

2. **Model Uncertainty (Epistemic Uncertainty):**

- Techniques such as Bayesian Neural Networks, dropout, or ensemble methods can be used to obtain estimates of model uncertainty along with predictions.
- For a deterministic Neural Network, we can train a network where after the pre-final layer we can attach two neural network layers that gives us the estimate of the mean and log-variance values of the logits. Using it we can construct a Target Distribution which is a multivariate Gaussian Distribution. The variance of the value across multiple values sampled from that distribution gives the estimate of Model Uncertainty.
- Bayesian neural networks replace the deterministic network's weight parameters with distributions over these parameters. Then we can sample network weights with which we can compute multiple output predictions corresponding to the set of each network weight. Epistemic uncertainty is estimated by taking the sample variance of the predictions from the sampled weights. Having lower sample variance means less epistemic uncertainty.
- We can also apply dropout during inference to get an ensemble of different models and the variance of the predictions across those values can give us an estimate of epistemic uncertainty.

REFERENCES:

1. A Survey of Uncertainty in Deep Neural Networks

[2107.03342] A Survey of Uncertainty in Deep Neural Networks (arxiv.org)

2. Predictive Uncertainty Estimation via Prior Networks

[1802.10501] Predictive Uncertainty Estimation via Prior Networks (arxiv.org)

3. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?

What Uncertainties Do We Need in Bayesian Deep Learning for...

4. Knowing known unknowns with deep neural networks.

Knowing known unknowns with deep neural networks