What Uncertainties tell you in Bayesian Neural Networks

Felix Laumann
Feb 15 · 9 min read ★

This time, we will examine what **homoscedastic, heteroscedastic, epistemic, and aleatoric uncertainties** actually tell you. In my opinion, this is an upcoming research field in Bayesian deep learning and has been greatly shaped by Yarin Gal's contributions. Most illustrations here are taken from his publications. But also see one of the field's latest contributions (https://arxiv.org/abs/1806.05978) where we propose a new, reliable and simple method how uncertainties should be computed.

As a background, in Bayesian deep learning, we have probability distributions over weights. Since most of the times we assume these probability distributions are Gaussians, we have a mean μ and a variance σ^2. The mean μ is the most probable value we sample for the weight.

> The variance can be seen as a measure of uncertainty — but what kind of uncertainty? Where is my neural network uncertain or what is my neural network uncertain about?

Basically, there are two groups of uncertainties and **the variance σ^2 is the sum of both**. We call them **aleatoric** and **epistemic** uncertainty. As we have mentioned in previous posts, we are interested in the predictive posterior probability distribution $p(y^*|x^*)$. But, this predictive distribution is intractable, unfortunately. What we need to do is approximating it by Laplace approximation (see this post for how to do it) and calculate the expected value, i.e. the mean μ, and the variance σ^2 of it.

$$\mathbb{E}_q[p(y^*|x^*)] = \int q_\theta(w|\mathcal{D})\ p_w(y^*|x^*)\ dw$$

$$\approx \frac{1}{T}\sum_{t=1}^{T} p_{w_t}(y^*|x^*)$$

Mean of predictive posterior probability distribution

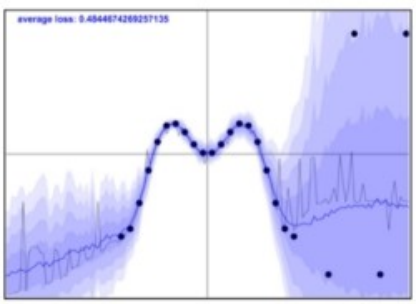
$$\mathrm{Var}_q\big(p(y^*|x^*)\big) = \text{aleatoric} + \text{epistemic}$$

Variance of predictive posterior probability distribution is the sum of aleatoric and epistemic uncertainty

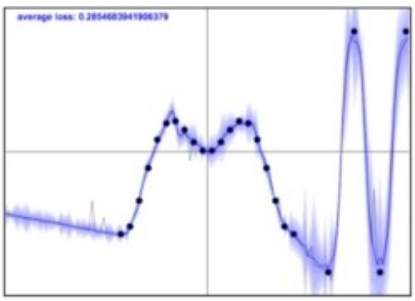
Let's now look in detail at both of these uncertainty estimations.

Aleatoric uncertainty

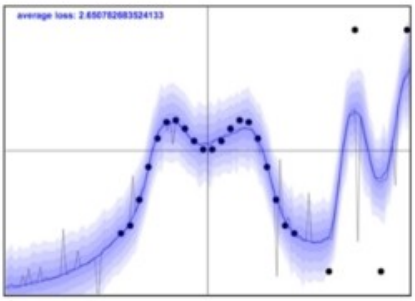
You have probably seen quite a few graphs which look similar to this one here below. We have a sample of observations, the black dots in the subsequent graphs which we assume to have *some* noise. If you had laboratory experiments in your chemistry, biology or physics high school lessons, you would know that no series of measurements ever is perfect. Especially when you measured a *y* value multiple times at the same point *x*, you rarely got the exact same *y*, didn't you? This is what we will call **aleatoric uncertainty**. Uncertainty about the observation *y* caused by a noisy data set *{x,y}*.



Heteroscedastic model with data-dependent observation noise.



Homoscedastic model with small observation noise.



Homoscedastic model with large observation noise.

Heteroscedastic aleatoric uncertainty

Exemplary, see the *x*-axis as a time scale from 8am to 10pm and we measure our heart rate over one week. We first take measurements in the morning at 8am just after getting up, another one at 10am after you have arrived in office after having cycled for 20 minutes and one in the evening at 6pm before you leave your office. Your heart rate at 8am is probably pretty steady over the week at around 80 beats per minute (bpm), perhaps ranging from 75 to 85 bpm. But, the measurements at 10am could vary from 120 to 160 bpm, depending how fast you cycled, in what shape you are every morning, etc. Then, in the evening when you had sat all day, your heart rate will be steady again around 90 bpm, perhaps ranging from 85 to 95 bpm.

> *What I just described is a real-world example of* ***heteroscedastic aleatoric uncertainty.*** *For every observation (x,y), we have a different extent of noise.*

Let us go a level higher and define this in mathematical terms as well. We can say that the output *y* of our deep learning model is sampled from a Gaussian distribution with a mean μ, which is the deterministic output $f(x)$ of the neural network dependent on the weights w, and a variance σ^2, which is also dependent on the weight w and the input x. Remind yourself, x is not the input vector of the entire data set, x is only *one* data point. We usually refer to the entire data set as X. Hence, we may have a different variance for every data point x, which we will recognise when we sample y multiple times, ranging widely.

$$y \sim \mathcal{N}(f^w(x), g^w(x)^{-1})$$

Homoscedastic aleatoric uncertainty

On the other hand, **homoscedastic** regression assumes identical observation noise for every input point x. Instead of having a variance being dependent on the input x, we must determine a so-called model precision τ and multiply it by the identity matrix I, such that all outputs y have the same variance and no co-variance among them exists. This model precision τ is the inverse observation standard deviation.

$$y \sim \mathcal{N}(f^w(x), \tau^{-1} I)$$

Epistemic uncertainty

Besides the uncertainty being caused by our somewhat noisy data, we can have other uncertainties, which we are actually able to minimise when we build our models. Epistemic uncertainty captures our ignorance about the models most suitable to explain our data. In other words, if the variance of our predictive distribution has a high epistemic uncertainty, you as a modeler know that you can do a much better job.

Let's keep our understanding for now on this level and explore how we can calculate these estimations.

Methods to compute aleatoric and epistemic uncertainty

How these two types of uncertainties are calculated is a upcoming research field, in my opinion. Here, I would like to discuss one specifically promising approach by Kwon et al (2018). Although it has made the estimation of these two uncertainties much easier, it has still its deficiencies, in my opinion.

Let us recap the basic algebraic formula for the variance:

$$\begin{aligned}\mathrm{Var}_q\big(p(y^*|x^*)\big) &= \mathbb{E}_q\big[(y - E[y])^2\big] \\ &= \mathbb{E}_q[yy^T] - \mathbb{E}_q[y]\mathbb{E}_q[y]^T\end{aligned}$$

The variance is the expected squared difference between any given output y and the expected value of any given input y.

This quantity can be decomposed into aleatoric and epistemic uncertainty:

$$\begin{aligned}\mathrm{Var}_q\big(p(y^*|x^*)\big) &= \mathbb{E}_q[yy^T] - \mathbb{E}_q[y]\mathbb{E}_q[y]^T \\ &= \int_\Omega \Big[\mathrm{diag}\Big(\mathbb{E}_{p(y^*|x^*,w)}[y^*]\Big) - \mathbb{E}_{p(y^*|x^*,w)}[y^*]\ \mathbb{E}_{p(y^*|x^*,w)}[y^*]^T\Big] q_\theta(w)\ dw \\ &+ \int_\Omega \Big[\mathbb{E}_{p(y^*|x^*,w)}[y^*] - \mathbb{E}_{q_\theta(y^*|x^*)}(y^*)\Big]\Big[\mathbb{E}_{p(y^*|x^*,w)}[y^*] - \mathbb{E}_{q_\theta(y^*|x^*)}(y^*)\Big]^T q_\theta(w)\ dw\end{aligned}$$

This equation is obtained from a variant of the law of the total variance. It is a big step and lots of mathematical reformulations need to be done here. Knowing Fubini's theorem is already very helpful, but the entire proof can be seen in Appendix A of Kwon's paper. I would recommend to go through it once, but don't bother too much, if you cannot follow all of the steps.

Let us go slowly through this equation to understand what is actually stands for. First of all, lets examine all the parameters and terms included:

- Ω is the space of all possible values for our weights w, denoted $w \in \Omega$.
- diag is the diagonal matrix. A diagonal matrix has entries along its diagonal and everywhere else zeros. If this diagonal matrix were the variance-covariance matrix of weights, we would have no covariances, only variances.
- $E[y^*]$ is the expected output of the input x^*. Carefully consider its disparate indices: we can have y^* based on the intractable predictive posterior distribution $p(y^*|x^*, w)$, or on the variational predictive posterior distribution $q(y^*|x^*, w)$ which has been optimised before for the parameters θ.
- $q(w)$ is the variational posterior distribution which approximates the intractable posterior distribution $p(w|D)$.

Aleatoric uncertainty

The first term of the predictive variance of the variational posterior distribution

$$\int_{\Omega} \left[\mathrm{diag}\left(\mathbb{E}_{p(y^*|x^*,w)}[y^*] \right) - \mathbb{E}_{p(y^*|x^*,w)}[y^*] \; \mathbb{E}_{p(y^*|x^*,w)}[y^*]^T \right] q_\theta(w) \; dw$$

is the **aleatoric uncertainty**. We first have the diagonal matrix of the expected outputs y^*, based on the intractable predictive posterior distribution $p(y^*|x^*, w)$. We subtract from it a matrix, which is the product of the expected outputs y^*, based on the intractable predictive posterior distribution $p(y^*|x^*, w)$ and its transpose. This entire construct is then multiplied with the variational posterior distribution and integrated over the weights w in the weight space Ω.

Before, we distinguished between heteroscedastic (different for each input) and homoscedastic (same for each input) aleatoric uncertainty. The term we use here can

be calculated per input (to obtain the heteroscedastic uncertainty) or as the average over all inputs (to obtain the homoscedastic uncertainty).

As you might have already guessed due to the integral, this term in intractable to estimate exactly. The most interesting part comes now: how do we estimate it? Kendall & Gal (2017) have proposed one simplifying method, but Kwon et al (2018) discuss this method's deficiencies for use with classifications and responds with another one:

$$\frac{1}{T}\sum_{t=1}^{T}\text{diag}(\hat{p}_t) - \hat{p}_t\,\hat{p}_t^T$$

$$\hat{p}_t = p(\hat{\omega}_t) = \text{Softmax}\{f^{\hat{\omega}_t}(x^*)\}$$

Let us follow their thought process to understand how they came up with such an estimator.

The central change is replacing

$$\mathbb{E}_{p(y^*|x^*,w)}[y^*]$$

with

$$\hat{p}_t = p(\hat{\omega}_t) = \text{Softmax}\{f^{\hat{\omega}_t}(x^*)\}$$

which we can do, because the Softmax-generated vector has probabilities as elements, hence computes the *variability* of the predictive distribution by repeating this calculation *T* times. The diagonal matrix is subtracted by another matrix, which is the Softmax-generated vector multiplied by its transpose.

Additionally, we take a sum instead of an integral to make it tractable, and instead of multiplying the sum with the variational posterior distribution, we calculate the average by dividing it all by *T*.

> *This gives us an average over the variability of the output coming from the data set. Hence, it can be seen as the uncertainty evolving from the variability of the data set.*

Epistemic uncertainty

The second term of the predictive variance of the variational posterior distribution

$$\int_{\Omega} \left[\mathbb{E}_{p(y^*|x^*,w)}[y^*] - \mathbb{E}_{q_\theta(y^*|x^*)}(y^*) \right] \left[\mathbb{E}_{p(y^*|x^*,w)}[y^*] - \mathbb{E}_{q_\theta(y^*|x^*)}(y^*) \right]^T q_\theta(w)\ dw$$

is the **epistemic uncertainty**. We have the exact same replacement as in the aleatoric uncertainty term, but an additional one is coming in.

$$\frac{1}{T} \sum_{t=1}^{T} (\hat{p}_t - \bar{p})(\hat{p}_t - \bar{p})^T$$

$$\text{where } \bar{p} = \sum_{t=1}^{T} \hat{p}_t / T \text{ and } \hat{p}_t = p(\hat{\omega}_t) = \text{Softmax}\{ f^{\hat{\omega}_t}(x^*) \}.$$

Here, we replace the expected outcome y^*, based on the variational distribution $q(y^*|x^*)$ with parameters θ, with the average of the Softmax-generated vectors of T samples. Afterwards, we subtract the Softmax-generated vectors by this average, and construct a matrix by multiplying this subtraction by its transpose.

Of course, we take again the sum instead of the integral to make it tractable.

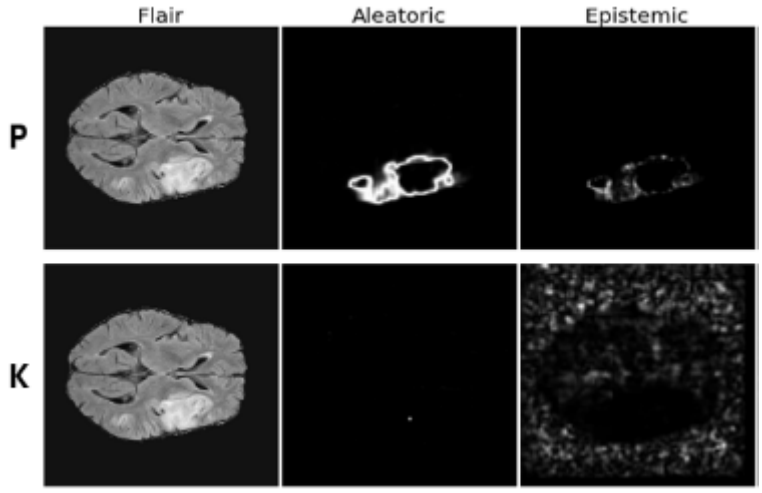
> *This gives us an average of the variability of the output coming from the model and is anti-proportional to the validation accuracy.*

Visualising results

These techniques are fairly simple to implement, especially for binary classification tasks in computer vision settings. Let us here examine one example of a biomedical image, precisely an MRI recording of a human brain.

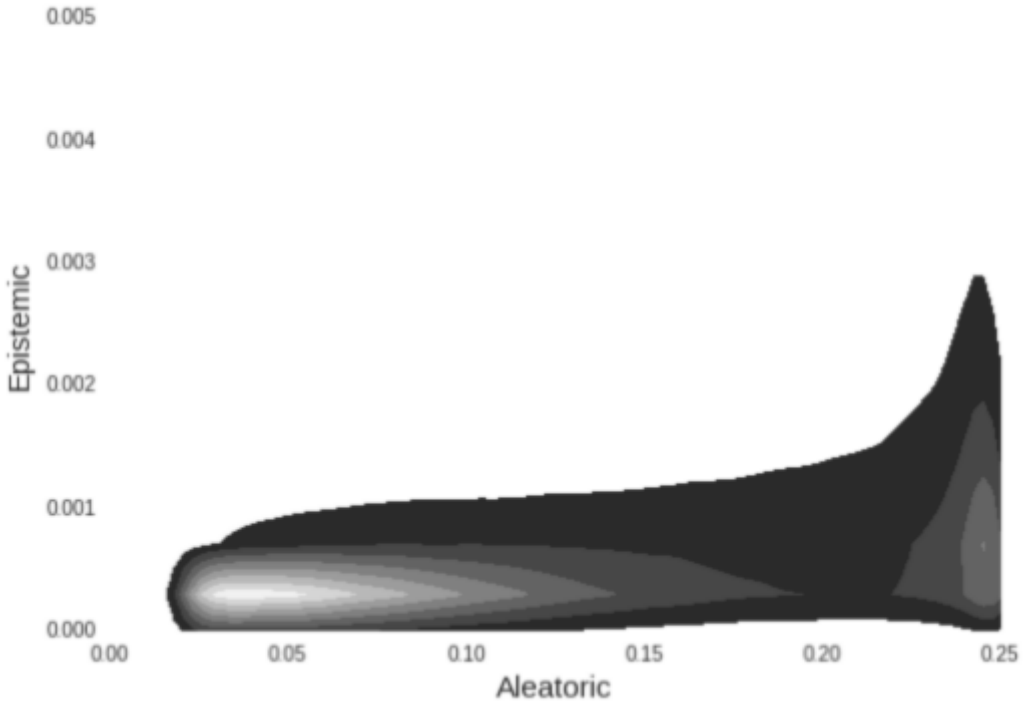
We calculated here the heteroscedastic aleatoric and epistemic uncertainty. Recall, heteroscedastic means a different uncertainty for every input. In image analyses, inputs

are the pixels. The figure below is a great comparison between the method by Kwon et al (2018) and the method by Kendall & Gal (2017).



P is the method by Kwon et al (2018), **K** is the method by Kendall & Gal (2017)

We can also compare the homoscedastic aleatoric and epistemic uncertainties of some set of data points against each other:



The brighter the area, the more data points are accumulated in this area.

We can also calculate the average homoscedastic aleatoric and epistemic uncertainties, but these numbers do not give us too many insights:

Aleatoric	7.6×10^{-4}	3.6×10^{-3}
Epistemic	2.2×10^{-5}	1.4×10^{-4}

For the data sets SISS and SPES

Softplus normalization

There have also been recent progress in the estimation of these uncertainties by a few colleagues and me: https://arxiv.org/abs/1806.05978

Here, we circumvent the bottleneck of implementing an additional Softmax function in the output layer by replacing it with the Softplus function and normalizing its output. We can write this uncertainty estimation then as:

$$\mathrm{Var}_q\big(p(y^*|x^*)\big) = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\mathrm{diag}(\hat{p}_t) - \hat{p}_t\,\hat{p}_t^T}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t-\bar{p})(\hat{p}_t-\bar{p})^T}_{\text{epistemic}}$$

where $\bar{p} = \frac{1}{T}\sum_{t=1}^{T}\hat{p}_t$ and $\hat{p}_t = \mathrm{Softplus_n}\big(f_{w_t}(x^*)\big)$.

The terrific consequence of doing this is that we have constant aleatoric uncertainties per data set, regardless of the model — exactly how it should be, because aleatoric uncertainties are purely dependent on the dataset. Nonetheless, this couldn't be achieved by previously published methods.

Another interesting outcome is the correlation between validation accuracy and epistemic uncertainty:

	Aleatoric uncertainty	Epistemic uncertainty	Validation accuracy
Bayesian VGG (MNIST)	0.00110	0.0004	99
Bayesian VGG (CIFAR-10)	0.00099	0.0013	85
Bayesian AlexNet (MNIST)	0.00110	0.0019	99
Bayesian AlexNet (CIFAR-10)	0.00099	0.0002	73
Bayesian LeNet-5 (MNIST)	0.00110	0.0026	98
Bayesian LeNet-5 (CIFAR-10)	0.00099	0.0404	69

With increasing validation accuracy, epistemic uncertainty decreases. This is logical: The more correct labels our model predicts, the more certain it is about these.

To see how we implemented this, please see our GitHub repo.

Final remarks

As previously said, this entire exploration of what the predictive variance is actually telling us is a greatly advancing research field and will give us lots of insights how our deep learning models can become better and better. Keep yourself updated with latest research in this field, these methods might be very helpful for your application since measures of uncertainty and measures for the origins of uncertainty are in any regard relevant when it comes to decision-making.

Machine Learning Artificial Intelligence Uncertainty Bayesian Statistics