Data-Driven Black-Box UQ for Boundary Value Problems

Grant Norman

4 April 2024

1 Problem

Let $u(x;\kappa)$ be the state of some system subject to an uncertainty $\kappa : \Omega \times \mathcal{D} \to \mathbb{R}$, where Ω is a sample space and with $x \in \mathcal{D}$ as the physical space. In other words, u is the state of some stochastic system, where the system has an underlying uncertain parameter κ. The goal of forward UQ is to propagate the distribution of κ through $u(x;\kappa)$ to find the distribution of u. However, we may not always fully know the distribution of κ. The inverse problem is determining the distribution of κ, possibly from observations of $u(x;\kappa)$.

Let us consider an example to demonstrate these ideas. Consider a variant of the homework problem, where

$$\frac{d}{dx}\left(\kappa\frac{du}{dx}\right) = 0, \quad u(0) = 0, \quad u(1) = 0, \tag{1}$$

where $\kappa = \exp\left(G(x;\omega)\right)$ and G has mean and covariance functions

$$\langle G(x)\rangle = 1, \quad C_{GG}(x_1,x_2) = \sigma^2 \exp\left(\frac{-|x_1-x_2|}{\ell}\right), \tag{2}$$

with $\ell, \sigma > 0$. *However, for this application, assume that κ is not known at all.* That is, we do not even know that κ is a log-normal random process. The inverse problem in UQ aims to characterize the distribution of κ, potentially given realizations for elementary event ω_i: $u(x;\kappa(\omega_i))$. Then, forward UQ seeks to propagate the distribution of κ to get the distribution of u. The distribution over functions u is difficult to work with, so instead we often simplify this problem to finding many samples taken from that distribution, or moments of that distribution. Note that if we have many samples, then we can also estimate the moments, but the converse is not necessarily true. Note, **this simple forward model allows for greater focus on the methods, which are the major contribution of this work**.

An alternative view removes the intermediate κ. Given samples $u(x;\kappa(\omega_i))$, our objective is to find the distribution giving these samples, or to generate more of those samples. The field of *generative modeling* takes this view, and seeks to generate new samples from that distribution. This view is necessary if the governing equations are unknown or very expensive to solve, or as an alternative to solving the inverse problem and then the forward problem separately.

In addition, we also consider a discretization of the functions $u(x;\kappa(\omega_i))$ as $\mathbf{u}$, for example as the value of u on an equispaced mesh, such as in finite differences. Then, our objective is to find $p(\mathbf{u})$, which we approximate through a number of samples.

2 Motivation

The overarching goal is to use some samples of a system to (cheaply) generate more samples of that system. More samples give a better picture of the uncertainty, although at the slow Monte Carlo convergence rate. Thus, it is often necessary to generate many samples. However, the original samples of the system themselves are often expensive to obtain, coming from experiments, or from expensive simulations. Based on these restrictions, the field of generative modeling proposes methods to generate many samples that follow the same distribution as the original samples, $p(\mathbf{u})$, by using samples already taken from that distribution.

In science and engineering, the samples are often measurements of a system governed by a Differential Equation (DE). Thus, a generative model would generate (discretized) solutions to the DE following $p(\mathbf{u})$. This provides cheap samples, as opposed to solving the DE directly or performing more experiments.

3 The Proposed Approaches

We hope to compare several generative models, which all aim to create samples $\mathbf{u}$ following $p(\mathbf{u})$. These approaches rely on already having some samples available, and their performance will be evaluated against a larger set of samples that are not used in their training.

3.1 Variational Autoencoder

First, following the course material, a Variational Autoencoder (VAE) will be used to generate samples of $p(\mathbf{u})$. VAEs extend autoencoders by encoding a useful, probabilistic structure on the latent space [1]. In other words, consistent with this notation, consider a latent vector $\mathbf{z}$, which we assume follows some nice distribution, typically a zero-mean multivariate normal distribution with an identity covariance matrix, $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$. The probabilistic encoder maps from an instance $\mathbf{u}$ to a latent space distribution, $p_\theta(\mathbf{z}|\mathbf{u})$. Similarly, the probabilistic decoder takes a latent space instance $\mathbf{z}$ and maps to a decoded distribution $p_\theta(\mathbf{u}|\mathbf{z})$, which ends up being a multivariate normal distribution. In the above, there is a brief change of notation, where the subscript denotes a parameterization of the distribution, for instance with θ parameterizing the joint distribution over $\mathbf{u}$ and $\mathbf{z}$. Computing the probabilistic encoder $p_\theta(\mathbf{z}|\mathbf{u})$ is theoretically done through Bayes Rule, but the denominator involves an integration over $\mathbf{u}$, which is intractable. Instead, $p_\theta(\mathbf{z}|\mathbf{u})$ is approximated through variational inference. Variational inference assumes a parameterized family of distributions and selects these parameters according to some error measurement, such as the Kullback-Leibler divergence. We denote this object $q_\phi(\mathbf{z}|\mathbf{u})$, which is also referred to as the probabilistic encoder, as it is what we will actually be using. We can take $q_\phi(\mathbf{z}|\mathbf{u}) = N(E_\phi(\mathbf{u}), \sigma_\phi(\mathbf{x})^2)$, where E_ϕ and σ_ϕ^2 parameterize the mean and variance respectively. In a VAE, the parameters ϕ are tuned through a gradient-based optimization with the Kullback-Leibler divergence (KL-divergence) as a proxy to "measure how far away" $q_\phi(\mathbf{z}|\mathbf{u})$ is from $p_\theta(\mathbf{z}|\mathbf{u})$. It is given as

$$D_{KL}(q_\phi(\mathbf{z}|\mathbf{u})||p_\theta(\mathbf{z}|\mathbf{u})) = \mathbb{E}_{z\sim q_\phi(\cdot|\mathbf{u})}\left[\log \frac{q_\phi(\mathbf{z}|\mathbf{u})}{p_\theta(\mathbf{z}|\mathbf{u}))}\right]. \tag{3}$$

Further simplifications then introduce the evidence lower bound (ELBO), which incorporates the likelihood $p_\theta(\mathbf{u})$, in addition to the KL-divergence. Maximizing the ELBO promotes matching the observed data, and making $q_\phi(\mathbf{z}|\mathbf{u})$ close to $p_\theta(\mathbf{z}|\mathbf{u})$.

3.2 Diffusion Model

Diffusion models are also generative models, related to the limit of a class of VAEs [2]. For now, they are described with less rigor than the aforementioned VAEs. They consist of a series of a series of invertible operations that transform from $\mathbf{z}$ towards $\mathbf{u}$, where again $\mathbf{z}$ is distributed according to some nice distribution. These also rely on the ELBO in their loss function.

3.3 Surrogate Dynamics

An alternative that makes use of the known structure of $\mathbf{u}$ is to assume that it is from an unknown boundary value problem. Perhaps the boundary conditions are known, but this is not a critical distinction. We assume some DE governs the system as

$$\mathcal{N}^\phi(x, u, u_x, u_{xx}, ...; \mathbf{z}) = 0, \tag{4}$$

where $\mathbf{z}$ again represents a latent variable following some nice distribution. This builds upon Raissi [3]. By directly solving $\mathcal{N}^\phi$ with a classical method, one arrives at an instance of $\mathbf{u}$ given an instance of $\mathbf{z}$.

We *hope* to show the following, starting with the academic problem described above.

1. This method can be trained in the same probabilistic way of deep learning and then used to generate samples consistent with the input data.
2. This method can be seen as a specific type of latent-space generative model, where the map from the latent-space to the decoded space involves a governing equation and its numerical solution.

3. By using more of the underlying structure common to science and engineering problems, this method outperforms the other generative modeling approaches, in terms of accuracy and number of samples required. However, we anticipate that this method will be more expensive at evaluation time.

4. Similarly, through the imposed structure, the model can then be extrapolated to unforeseen configurations (changing the geometry or the boundary conditions) and still produce meaningful results, unlike the other generative models.

In the case that this completely fails, more focus can be given to specializing the other established generative models to this problem, for instance with β-VAEs or with neural network architectures corresponding to the problem's local nature.

References

[1] Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. arXiv:1312.6114 [cs, stat]. Dec. 2022. URL: `http://arxiv.org/abs/1312.6114` (visited on 04/05/2024).

[2] Chin-Wei Huang, Jae Hyun Lim, and Aaron Courville. "A Variational Perspective on Diffusion-Based Generative Models and Score Matching". In: (). arXiv: 2106.02808v2. (Visited on 05/23/2023).

[3] Maziar Raissi. *Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations*. Tech. rep. arXiv: 1801.06637v1. 2018. URL: `https://github.com/maziarraissi/DeepHPMs` (visited on 06/20/2022).