Joakim Lunde

Solving Laplace's equation with Physics-Informed Neural Networks

TBM4500 - Geotechnics, Project Thesis
Supervisor: Ivan Depina
December 2024

Norwegian University of Science and Technology
Faculty of Engineering Science
Department of Civil and Environmental Engineering

NTNU

Abstract

The `ntnuthesis` document class is a customised version of the standard LaTeX `report` document class. It can be used for theses at all levels – bachelor, master and PhD – and is available in English (British and American) and Norwegian (Bokmål and Nynorsk). This document is ment to serve (i) as a description of the document class, (ii) as an example of how to use it, and (iii) as a thesis template.

Sammendrag

Dokumentklassen `ntnuthesis` er en tilpasset versjon av LaTeX' standard `report`-klasse. Den er tilrettelagt for avhandlinger på alle nivåer – bachelor, master og PhD – og er tilgjengelig på både norsk (bokmål og nynorsk) og engelsk (britisk og amerikansk). Dette dokumentet er ment å tjene (i) som en beskrivelse av dokumentklassen, (ii) som et eksempel på bruken av den, og (iii) som en mal for avhandlingen.

Zhang et al., 2022

Contents

List of Figures

List of Tables

Chapter 1

Literature study

1.1 Physics-informed neural networks for consolidation of soils

file:///C:/Users/jolu2/AppData/Local/Microsoft/Windows/INetCache/IE/79SHIDHO/Physics-informed_neural_networ[1].pdf

For the forward problem, it is difficult to obtain analytical solutions for most of the models related to consolidation.

Researchers have revealed that PINNs possess the following advantages compared with the conventional mesh-based numerical methods in tackling the forward problem. First, PINNs is capable of solving the inverse problem with the only minor change of the code that is used in a forward problem (Liu and Wang, 2019). Secondly, neural network-based methods with mesh-free features can reduce the tedious work of mesh generation (Basir and Senocak, 2022). Thirdly, PINNs can obtain remarkably accurate solutions and reliable parameter estimations with fewer data and average-quality data, to reduce the dependence on the need for large training datasets (Zhang et al., 2021). Fourthly, PINNs can produce results at any point in the domain once it has been trained (Basir and Senocak, 2022).

Uses Tanh

The values of wf and wb are commonly assumed to be 1 in the case that all the loss values are of the same order of magnitude. It should be noted that the choice of these two coefficients is still an open problem needing further investigations (Wang et al., 2021).

We use the Adam combined with L-BFGS optimizers

Furthermore, we employ a commonly used Glorot normal scheme (Glorot and Bengio, 2010) as the p

To deal with the stochastic nature of the training procedure, we calculated the results as an average over 5 realizations as suggested by Kadeethum et al. (2020)

Found that more with a lot of domain points, increasing the boundary points help, but not the other way around necessarily.- overfitting?

It is clear from Figure 7a that the performance of PINNs is found to be optimal at learning rates of 10–2 and 10–3

1.2 Probabilistic physics-informed neural network for seismic petrophysical inversion

https://library.seg.org/doi/10.1190/geo2023-0214.1?utm_source=chatgpt.com

We propose a probabilistic physics-informed neural network (P-PINN) algorithm for seismic petrophysical inversion with the goal of estimating the most likely model of petrophysical properties, the unknown hyperparameters of the rock-physics model, and the uncertainty of the model predictions.

1.3 Machine Learning-Aided Monte Carlo Simulation and Subset Simulation

https://journals.sagepub.com/doi/10.1177/03611981241248166?utm_source=chatgpt.com#body-ref-bibr14-03611981241248166

(...) geotechnical structure design and analysis are also affected by measurement uncertainty, statistical uncertainty, and transformation model uncertainty. These geotechnical uncertainties are not capable of being taken into account by conventional deterministic analytical methods. First of all, there are many different causes of uncertainty, including variations in soil characteristics, building techniques, and environmental factors (13).

1.4 Bayesian neural networks

1.4.1 Simple idea

The difference between a normal neural network and a Bayesian neural network is that the latter has a distribution of weights, rather than a single value. This allows for the model to express uncertainty in the predictions.

We have a KL term as well as the likelihood term in the loss function. The KL term is a measure of how much the posterior distribution differs from the prior distribution. The likelihood term is the negative log likelihood of the data given the weights. Posterior distribution, P(W|D), is the distribution of the weights given the data, and the prior distribution, P(W), is the distribution of the weights before seeing the data.

$$P(W|D) = \frac{P(D|W)P(W)}{P(D)} \tag{1.1}$$

We approximate $P(W|D)$ with a Gaussian distribution, and the KL divergence describes the difference between this distribution and the posterior distribution.

Another approach is to use Markov Chain Monte Carlo. Then we use a Monte Carlo instead of a simple distribution as replacement. Less efficient.

1.4.2 In Practice

Each weight have a distribution $W \sim \mathcal{N}(\mu, \sigma^2)$
Than compute the likelihood of

A numerical example:

NN(x;w,b) = w*x + b

$x_1 = 1$, $y_1 = 1.8$ $x_2 = 3$, $y_2 = 2.7$

We have a Gaussian prior distribution of the weights, $w = b = \mathcal{N}(0, 1^2)$

The data likelihood for a single data point is

$$p(y_i|w, b) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y_i - \text{NN}(x_i; w, b))^2\right) \tag{1.2}$$

We use variational inference to approximate the posterior distribution. q(w,b) = q(w)q(b). We use a Gaussian distribution for each weight.

$$q(w) = \mathcal{N}(\mu_w, \sigma_w^2) \tag{1.3}$$
$$q(b) = \mathcal{N}(\mu_b, \sigma_b^2) \tag{1.4}$$

Evidence lower bound (ELBO) is the objective function we want to maximize. It is written as: $ELBO = E[\log p(D|w, b)] + \ln(p(w, b)) - \ln(q(w, b))$, where the KL divergence is the difference between the prior and the posterior: $KL(q(w, b)||p(w, b)) = -\ln(q(w, b)) + \ln(p(w, b))$. The prior is written as $p(w, b) = p(w)p(b)$.

We fist sample from the prior distribution, then we compute the likelihood of the data given the weights, and then we compute the KL divergence between the prior and the posterior:

The prior distribution is $w = b = \mathcal{N}(0, 1^2)$. Let's say we draw w' = 0.37 and b' = -0.11 from the prior distribution. Then we compute the likelihood of the data given these weights:

$ELBO = E[\log p(D|w', b')] + \ln(p(w', b')) - \ln(q(w', b'))$

First term is the likelihood of the data given the weights. We have two data points, so we compute the log likelihood for each data point and sum them up:

$$\log p(D|w', b') = \log p(y_1|w', b') + \log p(y_2|w', b') \tag{1.5}$$
$$= \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y_1 - \text{NN}(x_1; w', b'))^2\right)\right) \tag{1.6}$$
$$+ \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y_2 - \text{NN}(x_2; w', b'))^2\right)\right) \tag{1.7}$$

Numerically, if we assume that the observation noise is gaussian with a standard deviation of 0.2, we get:

$$\log p(D|w', b') = \log\left(\frac{1}{\sqrt{2\pi}\cdot 0.2}\exp\left(-\frac{1}{2\cdot 0.2^2}(1.8 - \mathrm{NN}(1; 0.37, -0.11))^2\right)\right) \tag{1.8}$$

$$+ \log\left(\frac{1}{\sqrt{2\pi}\cdot 0.2}\exp\left(-\frac{1}{2\cdot 0.2^2}(2.7 - \mathrm{NN}(3; 0.37, -0.11))^2\right)\right) \tag{1.9}$$

When computed numerically, we get log likelihood = -64,38.

Next term is the log prior, $\ln(p(w', b')) = \ln(p(w'))+\ln(p(b')) = \ln\left(\frac{1}{2\pi}e^{-\frac{0.37^2}{2}}\right)+$ $\ln\left(\frac{1}{2\pi}e^{-\frac{0.11^2}{2}}\right)$

This becomes -0.988 - 0.926 = -1.914.

The last term is the log of the variational distribution, $\ln(q(w', b')) = \ln(q(w'))+$ $\ln(q(b')) = \ln\left(\frac{1}{2\pi}e^{-\frac{\mu_w^2}{2\sigma_w^2}}\right) + \ln\left(\frac{1}{2\pi}e^{-\frac{\mu_b^2}{2\sigma_b^2}}\right)$

1.5 Weight Uncertainty in Neural Networks

`https://arxiv.org/pdf/1505.05424`

Bayes by Backprop is just a type of inference method.

"In general, exact Bayesian inference on the weights of a neural network is intractable as the number of parameters is very large and the functional form of a neural network does not lend itself to exact integration."
In BNN, we write p(y|x,w) as the likelihood instead of the general p(D|W).

1.6 from chatGPT

An alternative to the variational inference is to use Markov Chain Monte Carlo (MCMC) methods. This is less efficient, but more accurate. VI: Faster, good for large-scale problems, but can suffer from approximation bias due to the factorized assumption. MCMC: More "accurate" in theory, but can be computationally heavy and slow to converge.

Real-World Analogy

Imagine you are trying to guess **how likely it is that a student will pass an exam**.

- You **don't know their intelligence** (w), but you have some prior belief about it.
- Given their intelligence, they have a certain probability of passing ($p(D \mid w)$).
- Since you **don't know their exact intelligence**, you average over all possibilities, weighted by how likely each intelligence level is.

$$p(\text{Pass}) = \int p(\text{Pass} \mid \text{IQ}) p(\text{IQ}) d(\text{IQ})$$

Figure 1.6.1: An analogy of $p(D) = \int p(D \mid w)p(w)dw$ explained by ChatGPT.

Bibliography

Zhang, Sheng et al. (June 2022). "Physics-informed neural networks for consolidation of soils". In: Engineering Computations. Accessed January 27, 2025. ISSN: 0264-4401. DOI: 10.1108/EC-06-2021-0255. URL: https://www.emerald.com/insight/content/doi/10.1108/ec-08-2021-0492/full/html.