Physics-consistency of infinite neural networks

Sascha Ranftl
Institute of Theoretical & Computational Physics, TU Graz
Petersgasse 16/II, 8010 Graz, Austria
ranftl@tugraz.at

Abstract

Recent research has highlighted the incorporation of prior physics knowledge into neural networks by construction of special neural activation functions. For this, we consider (i) Gaussian process kernels that adhere to the principles of physics, and (ii) the infinite-width correspondence between neural networks and Gaussian processes. Together, this begs the question for infinite-width neural networks that are consistent with the laws of physics. So construed regression models may find specialized applications such as inverse problems, uncertainty quantification or optimization of or with physical models. These 'surrogate' models should learn efficiently from limited data while generalizing in a physically sensible manner.

1 Background

The integration of prior physics knowledge into machine learning (ML) models has received increasing attention in the natural and computational sciences due to the need for 'surrogate models' that can perform with limited data, whilst they ought to give physically sensible and trustworthy predictions nevertheless. The "infinite-width correspondence" herein refers to the fact that neural networks (NNs) converge in the infinite-width limit towards Gaussian processes (GPs) [17]. This infinite-width limit has led to a number of key insights into NN deep learning and training dynamics such as the Neural Tangent Kernel (NTK) [11], or that Dropout is a Bayesian approximation [8]. On the other hand, the emerging paradigm of physics-informed ML (PIML) [13, 18, 19] had great impact in Computational Science & Engineering (CSE) by enabling high-dimensional surrogate models [25, 24]. In CSE, a surrogate model is a fast, parametrized approximation to complex, expensive large-scale simulations. A surrogate then enables many-query problems such as optimization, calibration or Uncertainty Quantification (UQ) [9, 14, 21]. The idea of PIML is essentially to regularize the training with a "physics-loss" (cf. Sec. 2.4) in order to achieve faster convergence and better generalization with less data.

While extremely useful once trained, PIML models are typically hard to train [27, 5, 7, 22]. We take a completely new approach to this problem based on a connection between GPs, NNs and physics in the form linear (differential) equations (DEs). Built on relationships in Fig. 1, the basics are provided by: (1) A set of methods to construct physics-consistent GP kernels for linear DEs [26, 15, 10, 1]. Non-linear Partial Differential Equations (PDEs) may potentially be feasible as well [4], but are not the subject of this work . (2) The fact that infinite (shallow or deep) NNs behave like GPs [17]. This is due to the Central Limit Theorem, which states that a large sum of random variables, e.g. a Bayesian neural network's output for any input, follows a Gaussian distribution under mild conditions (bounded activations, existing momenta). Note that these findings hold for a very general class of (deep) NNs, even e.g. Transformers [29, 16, 6]. (3) Mercer's theorem or the "kernel trick" relates GP kernels to their feature maps or basis functions. In reproducing kernel Hilbert spaces (RKHS), a kernel can be represented by inner products of its eigenfunctions, and vice versa functions can be represented by linear kernel combinations (representer theorems). Quoting literally from [12]: "Which activation function should we use? This is one of the most basic and

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

meaningful questions that could be posed. [...] there is no rule of thumb for choosing the optimal activation function". There are no principles beyond taxonomy or heuristics [3].

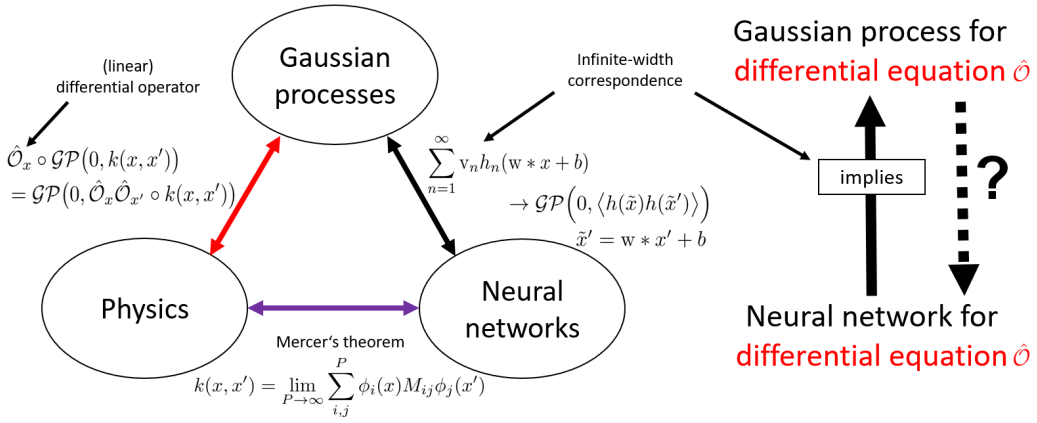

Figure 1: Overview of the basic relationships that connect the three mathematical notions of neural networks (NNs), Gaussian processes (GPs) and physical laws in the form of (differential) equations, here expressed as a (symbolic) operator $\hat{\mathcal{O}}_x$ acting on x. An example for such a symbolic operator would be $\hat{\mathcal{O}}_x \triangleq \nabla^2$, then $\hat{\mathcal{O}}_x f(x) := \nabla^2 f(x) = 0$ would imply that $f(x)$ obeys this homogeneous Laplace equation. (1) Linear operators $\hat{\mathcal{O}}$ applied to a GP $\mathcal{GP}$ with kernel k yield again a GP with a new kernel $\hat{\mathcal{O}}_x \hat{\mathcal{O}}_{x'} \circ k$. This can be seen e.g. from applying a linear operator to a (generalized) linear model first, and only in the second step applying the kernel trick. Cf. excellent literature [15]. (2) NNs in the large-width limit behave like GPs. (3) The "kernel trick"is based on Mercer's theorem.

2 Physics-consistent Gaussian processes & their Infinite Neural Networks

We introduce two *sets* of functions $V, W : X \to Y \subseteq \mathbb{R}^Q$. From the first *set* V, we denote two particular elements as $g(x) \in V,\ x \mapsto g(x)$ and $f(x) \in V,\ x \mapsto f(x), x \in X \subseteq \mathbb{R}^D$, where D, Q denote input/output dimension. No special assumptions are made on these (integer, finite) dimensions. We aim to learn these functions from data related to some physical law expressed in terms of a linear operator $\hat{\mathcal{O}} : V \to W, \hat{\mathcal{O}} \circ f(x) \mapsto \tilde{f}(x)$. In words, an operator maps a function from one set V to another set W. Linearity requires $\hat{\mathcal{O}} \circ (\alpha \cdot f + \beta \cdot g) = \alpha \cdot \hat{\mathcal{O}} \circ f + \beta \cdot \hat{\mathcal{O}} \circ g$ for some constants $\alpha, \beta \in \mathbb{R}$. Let the physical law be expressed in the form $\hat{\mathcal{O}} f(x) = 0$ ($\circ$ is omitted from now). E.g. $\hat{\mathcal{O}}$ could be the D-dimensional Laplace-operator $\hat{\mathcal{O}} \triangleq \nabla^2 := \sum_{k=1}^{D} \frac{\partial^2}{\partial x_k^2}$, x_k denoting the $k-th$ element of D-dimensional vector $x = (x_1, ..., x_D)^T$. In the following, g denotes a GP and f denotes a NN.

2.1 Physics-consistent Gaussian processes

Following closely the notation from [15], let g be a GP with mean function $\mu : \mathbb{R}^D \to \mathbb{R}^Q : x \mapsto \langle g(x) \rangle$ and a positive semi-definite covariance function $k : \mathbb{R}^D \oplus \mathbb{R}^D \mapsto \mathbb{R}^{Q \times Q}_{\succeq 0} : (x, x') \mapsto \left\langle \big(g(x) - \mu(x)\big) \big(g(x') - \mu(x')\big)^T \right\rangle$, where $\langle \cdot \rangle$ denotes expectations. For simplicity, $\mu \equiv 0$.

$$g(x) \sim \mathcal{GP}\big(0, k(x,x')\big) \tag{1}$$

We further require that g be consistent with a physical law expressed in terms of a *linear* differential operator $\hat{\mathcal{O}}_x$, i.e. differential wrt x, such that it admits the following equation

$$\hat{\mathcal{O}}_x g(x) = 0\ , \tag{2}$$

In other words, we use a GP as an ansatz for solving eq. (2). As mentioned before, this can be a differential operator defining a differential equation. If $g(x)$ is a GP, then, due to linearity of $\hat{\mathcal{O}}$, $\hat{\mathcal{O}}_x g(x)$ yields again a GP,

$$\hat{\mathcal{O}}_x g(x) \sim \mathcal{GP}\big(0, \hat{\mathcal{O}}_x k(x,x') \hat{\mathcal{O}}^\dagger_{x'}\big) \tag{3}$$

where † denotes the adjoint, acting on the *second* argument of the kernel function x' from the right [15]. We have introduced the subscript x to clarify on which argument of the kernel the differential operator is acting upon. The above relationship can be used to formulate a number of equivalent physics-consistency conditions for the GP (or its kernel, respectively), e.g. from [26]

$$\hat{\mathcal{O}}_x \hat{\mathcal{O}}_{x'} \; k(x,x') \Big|_{x=x'} \stackrel{!}{=} 0 \; . \tag{4}$$

The solution for k in eq. (4) then defines a GP that satisfies eq. (2) a priori, i.e. *before* training. In other words, these physics-consistency conditions can be used to find and design physics-consistent kernels. For this, Mercer's theorem can be especially useful, in that it states that a kernel function (i.e. an element from a reproducing kernel Hilbert space) can always be expressed in terms of a set of basis functions $\{\phi_i\}$ and vice versa [23]. Formally, this means

$$k(x,x') = \lim_{P \to \infty} \sum_{i,j}^{P} \phi_i(x) M_{ij} \phi_j(x') \tag{5}$$

with a suitable prior covariance matrix M for the function weights. Note that Green's functions or fundamental solutions are especially interesting candidates for basis functions, as their kernel inherits their physical properties [1, 20] (in a sense, applying the kernel trick to Green's functions).

2.2 Infinite neural networks

Let us now consider a NN f with a *single* hidden layer with N neurons with non-linear but bounded activation functions h, where the index $n = 1, ..., N$ sums over the N neurons in the hidden layer. w_n and v_n are the input-to-latent and latent-to-output weights respectively, and b are bias terms.

$$f(x) = \sum_{n=1}^{N} \mathrm{v}_n h_n \Big(\mathrm{w}_n \cdot x + b_n \Big) \; , \tag{6}$$

Under the infinite-width limit, $N \to \infty$, eq. (6) converges to a Gaussian process, i.e. a Gaussian distribution $\forall x$, by the Central Limit Theorem if the activation functions h_n are bounded and weights and biases are i.i.d. [17]. Note that a significantly larger class of *deep* NNs show this property, cf. [29]. The NN converges to a GP with kernel k, with following condition for equivalence

$$k(x,x') \stackrel{!}{=} \big\langle f(x) f(x') \big\rangle \; , \tag{7}$$

The integral r.h.s. can be solved numerically, and in some cases analytically [28]. Note that eq. (7) is essentially determined by the covariance of the activations for a fixed NN feed-forward architecture.

2.3 Physics-consistency condition for neural networks

Substituting the NN-GP consistency condition eq. (7) into the physics-consistency condition for the GP eq. (4) (or variants thereof), allows us to formulate a physics-consistency condition for the NN,

$$\hat{\mathcal{O}}_x \hat{\mathcal{O}}_{x'} k(x,x') \stackrel{!}{=} \big\langle \hat{\mathcal{O}}_x f(x) \; \hat{\mathcal{O}}_{x'} f(x') \big\rangle \; , \tag{8}$$

where pointy brackets denote expectation w.r.t. to NN parameters. This gives rise to the problem of finding pairs of probability density functions $p(\theta)$ for weights and biases and activations h such that the physics-consistency condition for the NN eq. (8) is satisfied. This implies that the choice of activation and prior cannot be simply treated separately. The choice of prior $p(\theta)$ and activation h in the infinite-width limit of eq. (6) is then analogous to the choice of the prior covariance matrix M and basis functions ϕ in the Mercer representation of the GP kernel in eq. (5). By applying the kernel trick to Green's functions, it follows from the choices $M_{ij} = \sigma_{\mathrm{v}_i} \delta_{ij}$ (Dirac-Delta), Gaussian priors, and the limit $p(\mathrm{w}_k) = \lim_{\sigma_{\mathrm{w}_k} \to 0} \mathcal{N}(0, \sigma^2_{\mathrm{w}_k} \Sigma)$ (where Σ is diagonal), that $h \to \phi$. That means, the activations become the Green's functions [20].

2.4 Training and regularization with physics-information

Given pairs of input-output data $D = \{x^{(d)}, y^{(d)}\}_{d=1}^{N_d}$, where $y^{(d)} = f(x^{(d)}) + \varepsilon$ are noisy observations, we ought to choose optimization criteria for the ML training parameters θ. E.g. here

$$\theta^* = \arg\min_{\theta} \left[\sum_{d=1}^{N_d} \Big(y^{(d)} - f(x^{(d)}) \Big)^2 + \lambda \sum_{p=1}^{N_p} \Big\| \Big(\hat{\mathcal{O}}_x f(x) \Big) \Big|_{x=x^{(p)}} \Big\| \right] , \tag{9}$$

which consists of the likelihood ("data term") and a "prior" ("physics-loss") defined via the residual of eq. (2). The last term denotes the differential operator $\hat{\mathcal{O}}$ to be applied to f and then evaluated at pivot points $x = x^{(p)}$ (which may be distinct from the data points $x^{(d)}$). This so-called "residual" is then minimized conjointly with the data mistfit. The choice of λ and $\{x^{(p)}\}_{p=1}^{N_p}$ is non-trivial. A NN trained this way is often called a Physics-Informed NN (PINN).

3 Experiment

We consider a Helmholtz equation

$$(\nabla^2 - c^2) f(x) = 0 \tag{10}$$

where $c = 6.0$ is a constant parameter related to the wavenumber. We restrict ourselves to 1D, i.e. $\nabla \equiv \frac{\partial^2}{\partial x^2}$. Then, the fundamental solution to this equation is a sinusoidal function. The physics-consistent GP kernel for this physical law in 1D has the form $k(x, x') = \cos\big(\alpha(x - x')\big)$ [2] with parameter $\alpha \in \mathbb{R}$. Since $\cos\big(\alpha(x - y)\big) = \sin(\alpha x)\sin(\alpha y) + \sin(\alpha x + \Delta)\sin(\alpha y + \Delta)$ with phase $\Delta \in \mathbb{R}$ [20], one particular activation that defines an infinite NN with the same physics-consistent covariance is a sinusoidal activation function, $h(\cdot) := \sin(\cdot)$. We learn NNs $f(x)$ from noisy observations (Gaussian, variance 0.04^2, homogeneous), and compare three distinct NNs: All three models are single-layer with the structure introduced before with $N = 100$ neurons in the hidden layer. Note the crucial assumptions are on the priors and not the likelihood. We denote as "vanilla" a simple NN with a sigmoid-activation, and training with L_2-loss. We denote as "physics-informed" the same network where the training is additionally regularized through a physics-loss term as in eq. (9). We compare the generalization behaviour of the three NNs in Fig. 2.

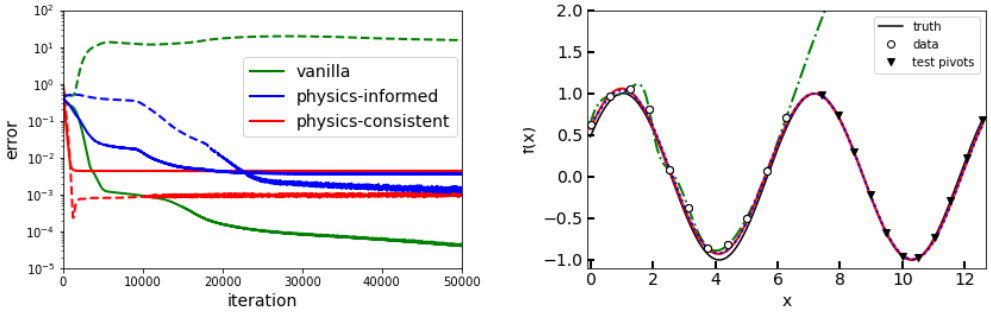

Figure 2: Plot of training (left, solid lines) and test error (dashed lines) convergence, and corresponding data & solutions (right). 'Vanilla' (green) refers to an ordinary single-layer NN with 100 neurons with sigmoid activations. 'Physics-informed' (blue) refers to the same NN but with additional training regularization by the usual corresponding physics-loss at the training (circle) and test (triangle) pivots (see eq. (9) as in [18]. 'Physics-consistent' (red) refers to again the same NN but with physics-based activations (sinus, cf. [20]) that has been derived from an infinite NN corresponding to its physics-consistent GP. We used ADAM with learning rate 10^{-3} and loss weight $\lambda = 1$. Clearly, a vanilla NN does not generalize well, is un-physical outside the data range and can overfit. In contrast, the PINN generalizes much better at the expense of training accuracy and convergence rate. Finally, the physics-consistent NN too generalizes much better, but converges about 1.5-2 orders of magnitude faster with less hyperparameter tuning. Error shown beyond iteration stopping for visualization. (right) Black, red & blue almost overlap.

4 Conclusion

Infinite NNs may provide a new avenue to incorporate prior physics knowledge into an NN's architecture via their corresponding physics-consistent GP. While the approach is generally non-trivial and yet limited to linear equations, the advantages in terms of convergence & generalization can be substantial in data-scarce problems. For future work, algorithmic approaches [15] to construct specialized, linearly constrained GP kernels may also be applicable to their infinite NN.

Acknowledgments

The author gratefully acknowledges funding by UFO - Unkonventionelle Forschung (Unconventional Research Program) of Das Land Steiermark (Styrian Government), Abt./Dept. 12 "Referat für Wissenschaft & Forschung", UFO-PN32.

References

[1] C. G. Albert. Gaussian processes for data fulfilling linear differential equations. *Proceedings*, 33(1):1–9, 2019. MaxEnt 2019 Garching/Germany.

[2] C. G. Albert. Physics-informed transfer path analysis with parameter estimation using Gaussian Processes. *International Congress on Acoustics*, (1):459–466, 2019.

[3] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, 2021.

[4] Y. Chen, B. Hosseini, H. Owhadi, and A. M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447:110668, 2021.

[5] S. Cuomo, V. S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):1–62, 2022.

[6] A. G. de G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In *ICML*, 2018.

[7] W. E and B. Yu. The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6:1–12, 2018.

[8] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

[9] R. Ghanem, D. Higdon, H. Owhadi, et al. *Handbook of uncertainty quantification*. Springer, 2017.

[10] M. Härkönen, M. Lange-Hegermann, and B. Raiţă. Gaussian process priors for systems of linear partial differential equations with constant coefficients. In *ICML*, 2023.

[11] A. Jacot, F. Gabriel, and C. Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *NeurIPS*, 2018.

[12] A. D. Jagtap and G. E. Karniadakis. How important are activation functions in regression and classification? A survey, performance comparison, and future directions. *Journal of Machine Learning for Modeling and Computing*, 4(1), 2022.

[13] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021.

[14] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[15] M. Lange-Hegermann. Algorithmic linearly constrained gaussian processes. In *NeurIPS*, 2018.

[16] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as Gaussian processes. In *ICLR*, 2018.

[17] R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1996. Chapter 2: Priors on infinite networks.

[18] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[19] M. Raissi, A. Yazdani, and G. E. Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

[20] S. Ranftl. A connection between probability, physics and neural networks. *Physical Sciences Forum*, 5(1), 2022. MaxEnt 2022.

[21] S. Ranftl and W. von der Linden. Bayesian Surrogate Analysis and Uncertainty Propagation. *Physical Sciences Forum*, 3(1). MaxEnt 2021.

[22] F. M. Rohrhofer, S. Posch, and B. C. Geiger. On the pareto front of physics-informed neural networks. *ArXiv: 2105.00862*, 2021.

[23] R. Schaback and H. Wendland. Kernel techniques: From machine learning to meshless methods. *Acta Numerica*, 15:543–639, 2006.

[24] R. Schöbi, B. Sudret, and J. Wiart. Polynomial-chaos-based Kriging. *International Journal for Uncertainty Quantification*, 5(2):171–193, 2015.

[25] R. K. Tripathy and I. Bilionis. Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of Computational Physics*, 375:565–588, 2018.

[26] K. G. van den Boogaart. Kriging for processes solving partial differential equations. In *Proceedings of the Conference of the International Association for Mathematical Geology (IAMG)*, 2001.

[27] S. Wang, X. Yu, and P. Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022.

[28] C. K. Williams. Computing with infinite networks. In *NeurIPS*, 1996.

[29] G. Yang. Tensor programs I: Wide feedforward or recurrent neural networks of any architecture are gaussian processes. In *NeurIPS*, 2019.