

A Connection between Probability, Physics and Neural Networks [†]

Sascha Ranftl 

Institute of Theoretical Physics-Computational Physics, Graz University of Technology, Petersgasse 16,
8010 Graz, Austria; ranftl@tugraz.at

[†] Presented at the 41st International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Paris, France, 18–22 July 2022.

Abstract: I illustrate an approach that can be exploited for constructing neural networks that a priori obey physical laws. We start with a simple single-layer neural network (NN) but refrain from choosing the activation functions yet. Under certain conditions and in the infinite-width limit, we may apply the central limit theorem, upon which the NN output becomes Gaussian. We may then investigate and manipulate the limit network by falling back on Gaussian process (GP) theory. It is observed that linear operators acting upon a GP again yield a GP. This also holds true for differential operators defining differential equations and describing physical laws. If we demand the GP, or equivalently the limit network, to obey the physical law, then this yields an equation for the covariance function or kernel of the GP, whose solution equivalently constrains the model to obey the physical law. The central limit theorem then suggests that NNs can be constructed to obey a physical law by choosing the activation functions such that they match a particular kernel in the infinite-width limit. The activation functions constructed in this way guarantee the NN to a priori obey the physics, up to the approximation error of non-infinite network width. Simple examples of the homogeneous 1D-Helmholtz equation are discussed and compared to naive kernels and activations.

Keywords: Bayes; probability; neural networks; Gaussian process; kernels; covariance functions; activation functions; physics-informed machine learning; differential equations; PDEs; ODEs; linear operators; linear constraints; inverse kernel trick



Citation: Ranftl, S. A Connection between Probability, Physics and Neural Networks. *Phys. Sci. Forum* **2022**, *5*, 11. <https://doi.org/10.3390/psf2022005011>

Academic Editors: Frédéric Barbaresco, Ali Mohammad-Djafari, Frank Nielsen and Martino Trassinelli

Published: 7 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

At the outset of this paper stands the observation that the remarkable success of neural networks (NNs) in the computer sciences is motivating more and more studies on their application in the natural and computational sciences. Such applications could be automatized lab data analysis, or the employment as a surrogate model in expensive many-query problems such as optimization or uncertainty quantification, see, e.g., [1,2]. Here, we desire not to employ ever larger off-the-shelf models, but to construct a simple NN and incorporate into its structure our prior knowledge of the physics. For this, we will recall two observations: First, that linear differential equations can a priori be incorporated into a Gaussian process (GP) model by proper choice of the covariance function, as was demonstrated by Albert [3] at the latest edition of this very meeting. Second comes the fact that GPs correspond to Bayesian NNs in an infinite-width limit, as demonstrated by Neal [4]. Both facts together imply that the physics can be incorporated a priori also into a NN model if one chooses the neural activation functions such that they match the corresponding GP's covariance function. The aim of this work is to introduce a formal notion of this connection between physics and NNs through the lens of probability theory, and demonstrate a so-derived principle for the construction of physical NNs with a simple example. It should be noted that this ansatz differs substantially from so-called “physics-informed” learning machines [5], where a regularization term for the optimization is introduced, but not incorporated into the structure.

2. Background and Related Work

A brief outline is given of the basic literature on Gaussian processes, (Bayesian) neural networks, and the infinite-width correspondence between the two. Related work on physically inspired GPs and NNs is discussed.

2.1. Gaussian Processes

Gaussian process regression or “Kriging” has been studied for decades [6]. Today, they are arguably the second-most popular class of machine learning models [7]. GPs have also been popular in the closely related field of “Uncertainty Quantification” [8,9]. A Gaussian process is essentially defined by its covariance function, often referred to as the kernel or in a wider sense correlation function. Thus, considerable work has been devoted towards the choice and design of such kernel functions [10].

It is known that a linear operator applied to a GP again yields a GP, but with a modified kernel [11]. This notion was also specialized to linear differential operators [12], and then applied to construct kernels for physical laws expressed in terms of so-defined differential equations, e.g., divergence-free fields [13], the Helmholtz equation [3] and the Poisson equation [14]; i.e., this resulted in GPs which, a priori, before training, are guaranteed to be divergence-free or obey the Helmholtz equation, respectively. The incorporation of other a priori knowledge such as boundary conditions was suggested by [15,16]. For an overview of more general linear constraints on GPs, see also [11].

The above mentioned approaches for constructing physical kernels [3,12] are fundamentally different from the popular “physics-informed GP” as popularized by [17] and earlier introduced by [15,18]. The physical GP differs from the physics-informed GP in the sense that the model is not merely “informed” of the physics post hoc through the training, but the model is constructed such that it obeys the physics a priori. The physics-informed approach consists in a GP as an ansatz function and regularizing the optimization by its residual. It is known that such a physics-informed GP generalizes better than their unregularized counterparts [5]. However, the physics-informed approach does not consider the form of the base kernel, the defining element of the GP, at all. Further, the physics-informed approach tolerates for the regressor to actually violate the physical requirements. The distinction of physical GPs from physics-informed GPs can also be understood as hard constraints in contrast to weak constraints on the model. It was also proposed to impose a GP prior on the inhomogeneity instead of the solution ansatz, and then construct the solution from the Green’s function [19,20].

2.2. Neural Networks

The arguably first-most popular class of machine learning models are NNs. Analogously to GPs and kernels, it is widely understood that the choice and design of activation functions is crucial for a NN. Many activations have been proposed, see, e.g., [10,21] for a recent and comprehensive overview. Ref. [22] proposed trigonometric activation functions, which by coincidence are very similar to the example that will be presented below. The design and choice of these activations are, however, seldomly guided by principles and methods, but rather trial and error. A small number of principled approaches through kernels, mostly relying again on the infinite-width correspondence, have been suggested [23–26], cf. Section 2.3. Vice versa, the infinite-width correspondence has also been used to compute kernels for GPs corresponding to particular activations.

The term “physics-informed neural network” was popularized by [27], building on the earlier work of [28,29]. Physics-informed NNs are very similar to physics-informed GPs in the sense that it serves as an ansatz function, the optimization of which is regularized by its residual. The “physics-informed” approach has been dominating due to its simplicity, easy implementation and wide applicability, despite suffering from the usual problems of NNs and inconsistencies in the multi-objective training [30]. In contrast, little attention was given to the “physics-constrained” approach, which aims at enforcing a hard constraint in contrast to a weak constraint implied by the physics-informed approach. For this, particularly the

enforcement of boundary conditions has been of special interest [31]. Additionally, the incorporation of symmetries has been investigated [32], e.g., rotation symmetries [23]. Comprehensive reviews on physics-informed learning have become available recently, yet little attention has been given to the kernels and activations [5,33].

2.3. The Infinite-Width Correspondence

As discovered by Neal [4], GPs and single-layer NNs are, under certain conditions and from a Bayesian point of view, equivalent in the infinite-width limit. Thus, Neal's insight allows to apply the theory of GPs to the analysis of NNs, and vice versa. It was later understood that Neal's theorem can be generalized to multi-layer networks [34], i.e., deep learning, and also holds after training [35] (i.e., with the posterior and not just the prior). This has led to a number of insights, e.g., that dropout regularization can be viewed as a particular prior in a Bayesian approximation [36]. Neal's theorem has also been generalized to convolutional NNs [37], and transformers or attention networks [38]. The infinite-width correspondence is proven to be relevant for many real world, large scale applications. Williams was supposedly the first to derive a GP kernel from the infinite-width limit of a single-layer NN with ReLU activations, giving rise to what many refer to as the "neural-net induced GP" [25]. Williams' result has also been generalized to multi-layer networks [23]. Others followed in deriving GP kernels from a number of activation functions [24,39]. Ref. [26] showed that various architectures and combinations of NNs correspond to combinations of GP kernels, e.g., multiplication of latent activations corresponds to a product-form of the kernel. Ref. [26] also derived a kernel corresponding to cosine functions as activation.

3. Method for Deriving Physical Activations

A formal notion of the infinite-width correspondence between GPs and NNs will be introduced, followed by procedures for the translation of linear differential constraints between GPs and NNs. For this, we will formally introduce GPs and review principles [3,12] for the design of physical kernels. Then, we will introduce a simple single-layer neural network and analyze the limit of infinite-width, providing the infinite-width correspondence of neural networks (NNs) to Gaussian processes (GPs). We will then show how Albert's principle for GPs is translated to NNs. We denote as $g(x)$ and $f(x)$ two learnable functions, which are governed by the same physical laws and depend on the same input, $x \in X \subseteq \mathbb{R}$ and $g, f : X \rightarrow Y \subseteq \mathbb{R}$. For g and f , we will assume a GP and a NN, respectively.

3.1. Gaussian Processes and Physical Kernels

We assume g to be a zero-mean GP,

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

where $k(x, x')$ is the to-be-determined covariance function or, synonymously, the kernel. An extension to non-zero mean functions should be straight forward. We further know a priori that g is governed by a physical law expressed in terms of the linear operator $\hat{\mathcal{O}}_x$ as

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

where $\hat{\mathcal{O}}_x$ is the operator defining the physical law in dependence on the variable x and acting on g through x . This operator could be a differential operator defining a differential equation, e.g., the Hamilton operator or more specifically the operator that defines the Helmholtz equation:

$$\hat{\mathcal{O}}_x \triangleq \nabla^2 + \nu^2, \quad (3)$$

with some constant wave number ν . Other examples would be $\hat{\mathcal{O}} \triangleq \nabla^2$, then this would be the Laplace equation, and would imply that g obeys this Laplace equation. It would also

imply that all measurements y , i.e., observed data of $g(x)$, namely $y = g(x) + \eta$, would obey this Laplace equation up to the measurement noise η . If the noise is independent of the input and the operator is linear, then $\hat{\mathcal{O}}y = \eta$. This latter fact could hypothetically inform us on the choice of the likelihood, but that is not the matter of this work.

Let us now find a kernel $k(x, x')$ such that Equation (2) is indeed fulfilled. It was shown by van den Boogaart [12] that if Equations (1) and (2) hold simultaneously, that this is equivalent to

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

That means, that if we find a kernel k such that Equation (4) is fulfilled, then the Gaussian process in Equation (1) defined by that kernel k also obeys Equation (2) and vice versa. It is nothing else than that we demand the GP to fulfill the PDE exactly, and the variance or uncertainty of the GP be zero at any particular probing location aside from a nugget that would model a measurement error. Note that Boogaart also showed this equivalence for non-zero mean functions. Mercer's theorem suggests a principled approach to solving Equation (4), in that it states that a kernel can be represented by a sum of products of basis functions ϕ [40],

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

with a suitable prior covariance matrix M for the function weights (not shown). It was argued by Albert [3] that Mercer's kernel can be built by considering the basis functions ϕ to be fundamental solutions to the differential equations, see Equation (2). Going from right to left in Equation (5) with a given basis may also be understood as "the kernel trick" [7], where the inner products of large bases are substituted with the corresponding kernel in order to avoid the computation of large matrices. We will later make an attempt at something like an "inverse" of this kernel trick, i.e., going from left to right and substituting the kernel through its basis function representation. Substitution of Equation (5) into Equation (4) yields

$$\lim_{P \rightarrow \infty} \sum_{i,j}^P \underbrace{\left(\hat{\mathcal{O}}_x \phi_i(x) \right)}_{=\psi_i(x)} M_{ij} \underbrace{\left(\hat{\mathcal{O}}_{x'} \phi_j(x') \right)}_{=\psi_j(x')} = 0, \quad (6)$$

where linearity was used and the operators only act on one term in the product each. One may argue that the operator acting upon a basis function ϕ should give rise to a new function ψ . It would be interesting under which circumstances this defines a new basis.

3.2. Neural Networks and the Infinite-Width Correspondence

Let us consider a NN f with a *single* hidden layer with N neurons with non-linear, but bounded activation functions h ,

$$f(x) = \sum_{k=1}^N v_k h_k(w_k \cdot x + a_k) + b, \quad (7)$$

where the index $k = 1, \dots, N$ sums over the N neurons in the single hidden layer. w_k are the input-to-latent weights, a_k and b are bias terms and v_k are the latent-to-output weights. Generalization to multiple features is straightforward through adding another index to the weights. In the infinite-width limit, $N \rightarrow \infty$, Equation (7) converges to a Gaussian process by the Central Limit Theorem if the activation functions h_k are all identical and bounded [4]. Let b and all v_k 's have zero-mean, independent Gaussian priors with variances σ_b^2 and $\sigma_{v_k}^2$, respectively. Let $\langle \cdot \rangle$ denote the expectation value with respect to all weights. Then

$\langle f(x) \cdot f(x') \rangle$ is the NN's output's covariance. A specific NN is equivalent to a specific GP with kernel $k(x, x')$ if the NN's covariance is equal to the kernel,

$$\begin{aligned} k(x, x') &\stackrel{!}{=} \langle f(x)f(x') \rangle \\ &= \sigma_b^2 + \lim_{N \rightarrow \infty} \sum_{k=1}^N \sigma_{v_k}^2 \langle h_k(x; \mathbf{w}_k) \cdot h_k(x'; \mathbf{w}_k) \rangle, \\ &= \sigma_b^2 + A^2 \langle h(x; \mathbf{w}) \cdot h(x'; \mathbf{w}) \rangle, \end{aligned} \quad (8)$$

where we used the linearity of the expectation in the first line. In the second line, the limit has been carried out under the assumption of identical neurons $h_k = h$ after fixing $\sigma_{v_k} = A/\sqrt{N}$ for some fixed A , as argued by Neal [4] (ch.2) or Williams [25]. This relationship has been used to compute GP kernels for given neural activation functions, e.g., a sigmoidal activation [25]. Note that it is crucial that the prior reflects that the hidden units are independent, as otherwise the Central Limit Theorem no longer holds. In order to design a NN that a priori obeys the physics, it would already be sufficient to find neural activations h such that Equation (8) holds for a particular GP kernel fulfilling Equation (4). We may, however, observe other possibly useful relationships.

3.3. Physical Neural Activation Functions

We may now insert the Mercer representation of the kernel, Equation (5), on the left hand side of the infinite-width correspondence, Equation (8),

$$\lim_{P \rightarrow \infty} \sum_{i,j}^P \phi_i(x) M_{ij} \phi_j(x') = \sigma_b^2 + \lim_{N \rightarrow \infty} \sum_{k=1}^N \sigma_{v_k}^2 \langle h_k(x; \mathbf{w}_k) \cdot h_k(x'; \mathbf{w}_k) \rangle. \quad (9)$$

Let the biases be known and fixed, i.e., $\sigma_b = 0$, then one solution to this equation is

$$M_{ij} = \sigma_{v_i} \delta_{ij}, \quad (10a)$$

$$p(\mathbf{w}_k) = \delta(\hat{\mathbf{w}}_k), \quad (10b)$$

$$h_m(x; \mathbf{w}_m) = \phi_m(x). \quad (10c)$$

The input-to-latent weights $\mathbf{w}_m$ now correspond to the implicit parametrizations of the GP basis functions ϕ_m . That would mean that the functions defining the Mercer representation of a GP-kernel can be particularly useful for finding the activation functions of the corresponding equivalent NN in the infinite-width limit if the weights' prior is chosen according to the Mercer representation. For some cases, there will be analytical solutions. We may now coerce the GP to obey some physical law $\hat{\mathcal{O}}$ and find

$$\lim_{P \rightarrow \infty} \sum_{i=1}^P \sum_{j=1}^P \psi_i(x) M_{ij} \psi_j(x') = \lim_{N \rightarrow \infty} \sum_{k=1}^N \langle \underbrace{\hat{\mathcal{O}}_x h_k(x; \mathbf{w})}_{=\tilde{h}_k(x; \mathbf{w})} \cdot \underbrace{\hat{\mathcal{O}}_{x'} h_k(x'; \mathbf{w})}_{=\tilde{h}_k(x'; \mathbf{w})} \rangle_{\mathbf{w}}. \quad (11)$$

The application of the differential operator to activation functions h yields new functions $\tilde{h}$. By finding the Mercer representation of a kernel, which coerces a GP to obey a physical law, we can find corresponding neural activation functions that are capable of forcing a NN to obey the same physical law, too.

3.4. Training

Given pairs of input–output data $D = \{x^{(d)}, y^{(d)}\}_{d=1}^{N_d}$, we ought to choose a criterion or loss function for the optimization. The optimal parameters w^* are defined by the loss function, here

$$w^* = \arg \min_w \left[\sum_{d=1}^{N_d} \left(y^{(d)} - f(x^{(d)}) \right)^2 + \lambda \sum_{p=1}^{N_p} \left\| \left(\hat{\mathcal{O}}_x f(x) \right) \Big|_{x=x^{(p)}} \right\| \right], \quad (12)$$

which consists of a ‘data loss’, i.e., the sum of squared errors, and a ‘physics-loss’, i.e., the residual of Equation (2) if network f is substituted as ansatz function, and a tuneable regularization parameter λ . The second term on the right hand side is to be understood as the differential operator applied to the function f , which is then evaluated at N_p pivot points $x^{(p)}$. A suitable norm $\|\cdot\|$ is to be chosen. For the proposed physics-constrained network, the physics-loss vanishes and λ can be arbitrary. For a vanilla neural network, λ would be set to zero. For a physics-informed network (see also [27]), a finite λ is chosen. The outcome is generally highly sensitive to the choice of λ , hence usually a hyperparameter optimization must be conducted.

4. Numerical Example

The approach shall be illustrated with the simple example of the one-dimensional homogeneous Helmholtz equation, i.e.,

$$\left(\frac{\partial^2}{\partial x^2} + \nu^2 \right) f(x) = 0. \quad (13)$$

The kernel derived from our first principles for the Helmholtz equation, referred to as the (1D-) Helmholtz kernel from hereon, has the following simple form [41]:

$$k(x, x') = \cos(\alpha(x - x')). \quad (14)$$

It should be noted that $\cos(\alpha(x - y)) = \sin(\alpha x) \sin(\alpha y) + \sin(\alpha x + \Delta) \sin(\alpha y + \Delta)$ with phase Δ . Following Equation (8) and under Equation (10), we may then choose sinusoidal activation functions

$$h(x) = \sin(x), \quad (15)$$

which correspondingly will be referred to as the (1D-) Helmholtz activation. The ‘frequency’ α and ‘phase’ Δ are then to be learned by the NN as weights w_k and biases a_k . We now proceed to illustrate the example numerically. The implementation is remarkably simple, and with high-level libraries such as PyTorch amounts circa to the re-writing of one line of code in a vanilla NN. For the following experiments, 11 noisy observations at equidistant pivot points from the fundamental solution to Equation (13) have been generated with a noise level of 20%, a frequency of $\omega = 0.51$ and $\phi = 0.50001$. The role of the choice of pivot points is not of primary concern here. As optimizer we choose ADAM [42] with a learning rate of 0.02. In the case of the physics-informed NN, the weight for the physics-loss was set to $\lambda = 0.1$. We note that experiments with stochastic gradient descent (various learning rates between 0.001 and 1.0) usually did not lead to sufficient convergence of the physics-informed NN (ReLU activations and training regularization, $\lambda \neq 0$) and the vanilla NN (ReLU activations and no training regularization, $\lambda = 0$) within a few thousand iterations, while the here constructed physical NN still converged (albeit slower). Figure 1 (right) presents the learned solutions to the Helmholtz equation, Equation (13), and Figure 1 (left) shows convergence plots. For reference, our physically-constructed NN is compared to a merely physics-informed NN (i.e., regularized training as discussed in Section 3.4) with ReLU activations as well as a plain vanilla NN with ReLU activations. Unsurprisingly, the

NN that has the physics incorporated into its structure and design vastly outperforms its merely informed or non-informed counterparts.

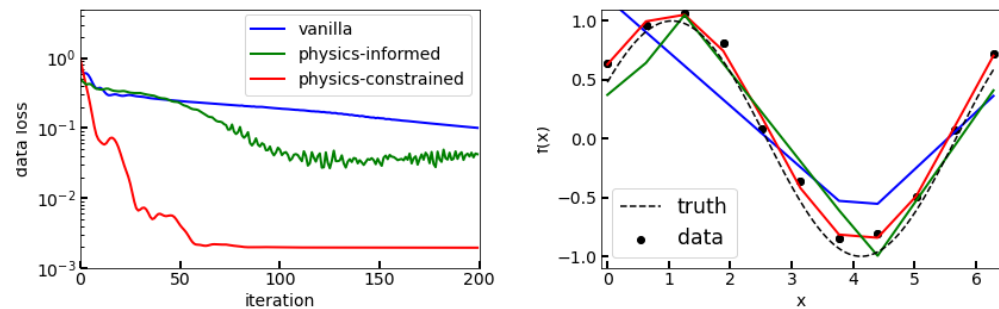


Figure 1. Plot of convergence (left) and corresponding solutions (right). For the visualization only, only the data loss is shown since the physics loss vanishes in the physics-constrained case. ‘Vanilla’ (blue) refers to an ordinary NN with ReLU activations. ‘Physics-informed’ (green) refers to the same NN, but with additional training regularization by the physics-loss, Equation (12). ‘Physics-constrained’ (red) refers to the same NN, but with our physics-based activations.

5. Discussion

In the above presented example, it turns out that the obtained activation function is also the fundamental solution to the differential equation, Equation (2). Since the kernel may be built from fundamental solutions as basis functions ϕ in Mercer’s representation, Equation (5), it seems that under the conditions of Equation (10) the neural activations h should also be proportional to those fundamental solutions. It may seem sensible to construct the solution from a combination of fundamental solutions, as it amounts to for this example. It should also be observed that our so-constructed NN also has the form of a Fourier series. Curiously, at the outset of the modern day perspective on regression through NNs, we have arrived at the conclusion that the proper physical NN assumes the simple forms that have been known all along. While this could be perceived as a circular journey, it seems reasonable that the result should be consistent with earlier perspectives on the same problem.

The here discussed connection can, in theory, be exploited to obtain a principled catalog of equivalences between physical laws and their corresponding kernel functions as well as activation functions. There are at least three conceivable procedures for finding solutions to Equation (11). One procedure would be to find matching examples in a forward manner, similar to [25], by defining neural activations by trial and error until they match a particular kernel. Any of the several representations from above may be used for that. This can be guesswork, if not less effective or necessarily less efficient. A more systematic approach would be through Mercer’s kernel representation, e.g., with fundamental solutions as basis functions as argued before, implying an “inverse kernel trick”. For many physical laws, this might be possible numerically only, if at all, which could be crucial for adoption of the procedure to more complex physical laws. It is also argued that it is possible to “skip” the construction of the kernel and directly find the activation function by demanding the right hand side of Equation (11) be zero. It should further be noted that Boogaart [12] also displayed equivalent formulations of Equation (4) and gives “four methods, too old to be found in the books I read”. These should be useful for constructing physical activations, too.

A limitation of the approach consists in the fact that our basic assumption were linear operators, i.e., non-linear equations cannot be treated in this way directly in the sense that non-linear operators applied to GPs do not yield GPs anymore. Hence, Equation (4) for constructing physical kernels does not hold anymore. As a consequence, no physical activations can be derived in the infinite-width correspondence either. A mitigation of this problem could lie in the linearization of the non-linear equations in order to retain the

original assumptions. For highly non-linear effects, such an approximation may however be problematic. Another limitation is that, in reality, NNs always have finite width. It remains unclear how wide a NN really needs to be for behaving approximately like a GP in this context.

We also seek to address the limitations of the experimental study. For one, no optimization of architecture details and hyperparameters has been carried out. ‘Vanilla’ NNs should be well capable of learning the Helmholtz equation if enough neurons are added, and if the optimization scheme is chosen appropriately. Particularly, the performance of the vanilla physics-informed NN depends strongly on the choice of the weighting hyperparameter λ in the multi-objective loss. It is well possible that, if chosen appropriately, such a NN might outperform. How an appropriate λ can be found efficiently, however, is an open problem to date. The claim of this paper is not that the here-constructed physical neural activation functions generally outperform physics-informed NNs, although it does in the above presented example. This work merely points out that the connection between GPs and NNs can be exploited for the principled design of physical neural activations based on linear differential equations. In contrast to a physics-informed NN, the physics-loss in Equation (12) rhs then vanishes a priori, and hence the tuning of the highly sensitive hyperparameter λ becomes obsolete.

Finally, many open questions remain, particularly also in anticipation of the various generalizations and other applications of the infinite-width correspondence, cf. Section 2.3. Can the approach be extended to deep NNs or even more complex architectures like LSTMs, Transformers or Graph Neural Networks? Do the assumptions still hold during training? A heuristic idea would be to simply replace the activations in the last hidden layer of a deep NN with physical activations. How can we include symmetries, boundary or initial conditions systematically, and are Green’s functions a viable path for that? Can the approach be beneficial for differential equations without known fundamental solutions?

Funding: The author was funded by Graz University of Technology (TUG) through the LEAD Project “Mechanics, Modeling, and Simulation of Aortic Dissection” (biomechaorta.tugraz.at) and supported by GCCE: Graz Center of Computational Engineering.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All information is contained in the manuscript.

Acknowledgments: The author would like to thank Wolfgang von der Linden, Thomas Pock, Christopher Albert and Robert Peharz for their helpful support and comments and the MaxEnt 2022 community for valuable discussions.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Ranftl, S.; von der Linden, W. Bayesian Surrogate Analysis and Uncertainty Propagation. *Phys. Sci. Forum* **2021**, *3*, 6. [\[CrossRef\]](#)
2. Ranftl, S.; Rolf-Pissarczyk, M.; Wolkerstorfer, G.; Pepe, A.; Egger, J.; von der Linden, W.; Holzapfel, G.A. Stochastic Modeling of Inhomogeneities in the Aortic Wall and Uncertainty Quantification using a Bayesian Encoder-Decoder Surrogate. *Comput. Methods Appl. Mech. Eng.* **2022**, *401 Pt B*, 115594. [\[CrossRef\]](#)
3. Albert, C.G. Gaussian processes for data fulfilling linear differential equations. *Proceedings* **2019**, *33*, 5. [\[CrossRef\]](#)
4. Neal, R.M. Bayesian Learning for Neural Networks. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 1996; Chapter 2: Priors on infinite networks. [\[CrossRef\]](#)
5. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422. [\[CrossRef\]](#)
6. O’Hagan, A. Curve Fitting and Optimal Design for Prediction. *J. R. Stat. Soc. Ser. B (Methodol.)* **1978**, *40*, 1–24. [\[CrossRef\]](#)
7. Rasmussen, C.E.; Williams, C.K. *Gaussian Processes for Machine Learning*; The MIT Press: Cambridge, MA, USA, 2006.
8. Bilonis, I.; Zabarar, N.; Konomi, B.A.; Lin, G. Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification. *J. Comput. Phys.* **2013**, *241*, 212–239. [\[CrossRef\]](#)
9. Schöbi, R.; Sudret, B.; Wiart, J. Polynomial-chaos-based Kriging. *Int. J. Uncertain. Quantif.* **2015**, *5*, 171–193. [\[CrossRef\]](#)

10. Duvenaud, D.K. Automatic Model Construction with Gaussian Processes. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2014. [\[CrossRef\]](#)
11. Swiler, L.P.; Gulian, M.; Frankel, A.L.; Safta, C.; Jakeman, J.D. A Survey of Constrained Gaussian Process Regression: Approaches and Implementation Challenges. *J. Mach. Learn. Model. Comput.* **2020**, *1*, 119–156. [\[CrossRef\]](#)
12. van den Boogaart, K.G. Kriging for processes solving partial differential equations. In Proceedings of the Conference of the International Association for Mathematical Geology (IAMG), Cancun, Mexico, 6–12 September 2001.
13. Jidling, C.; Wahlstrom, N.; Wills, A.; Schön, T.B. Linearly Constrained Gaussian Processes. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 1215–1224. [\[CrossRef\]](#)
14. Dong, A. Kriging Variables that Satisfy the Partial Differential Equation $\Delta Z = Y$. In Proceedings of the Geostatistics. Quantitative Geology and Geostatistics; Armstrong, M., Ed.; Springer: Dordrecht, The Netherlands, 1989; Volume 4, pp. 237–248. [\[CrossRef\]](#)
15. Graepel, T. Solving Noisy Linear Operator Equations by Gaussian Processes: Application to Ordinary and Partial Differential Equations. In Proceedings of the 20th International Conference on International Conference on Machine Learning, ICML'03, Washington, DC, USA, 21–24 August 2003; pp. 234–241. [\[CrossRef\]](#)
16. Gulian, M.; Frankel, A.; Swiler, L. Gaussian process regression constrained by boundary value problems. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114117. [\[CrossRef\]](#)
17. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Machine learning of linear differential equations using Gaussian processes. *J. Comput. Phys.* **2017**, *348*, 683–693. [\[CrossRef\]](#)
18. Särkkä, S. Linear Operators and Stochastic Partial Differential Equations in Gaussian Process Regression. In Proceedings of the Artificial Neural Networks and Machine Learning—Proceedings of the 21st International Conference on Artificial Neural Networks, ICANN'11, Espoo, Finland, 14–17 June 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 151–158. [\[CrossRef\]](#)
19. Álvarez, M.A.; Luengo, D.; Lawrence, N.D. Linear latent force models using gaussian processes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2693–2705. [\[CrossRef\]](#) [\[PubMed\]](#)
20. López-Lopera, A.F.; Durrande, N.; Álvarez, M.A. Physically-inspired Gaussian process models for post-transcriptional regulation in Drosophila. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *18*, 656–666. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Apicella, A.; Donnarumma, F.; Isgrò, F.; Prevete, R. A survey on modern trainable activation functions. *Neural Netw.* **2021**, *138*, 14–32. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Ngom, M.; Marin, O. Fourier neural networks as function approximators and differential equation solvers. *Stat. Anal. Data Min.* **2021**, *14*, 647–661. [\[CrossRef\]](#)
23. Tsuchida, R.; Roosta, F.; Gallagher, M. Invariance of Weight Distributions in Rectified MLPs. In Proceedings of the 35th International Conference on Machine Learning (ICML'18), Stockholm, Sweden, 10–15 July 2018; Volume 80 pp. 4995–5004. [\[CrossRef\]](#)
24. Cho, Y.; Saul, L.K. Kernel Methods for Deep Learning. In Proceedings of the 22nd International Conference on Neural Information Processing Systems, NIPS'09, Vancouver, BC, Canada, 7–10 December 2009; Curran Associates Inc.: Red Hook, NY, USA, 2009; pp. 342–350. [\[CrossRef\]](#)
25. Williams, C.K.I. Computing with Infinite Networks. In Proceedings of the 9th International Conference on Neural Information Processing Systems, NIPS'96, Denver, CO, USA, 2–5 December 1996; MIT Press: Cambridge, MA, USA, 1996; pp. 295–301. [\[CrossRef\]](#)
26. Pearce, T.; Tsuchida, R.; Zaki, M.; Brintrup, A.; Neely, A. Expressive Priors in Bayesian Neural Networks: Kernel Combinations and Periodic Functions. In Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI), Tel Aviv, Israel, 22–25 July 2019; p. 25. [\[CrossRef\]](#)
27. Raissi, M.; Perdikaris, P.; Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
28. Lagaris, I.E.; Likas, A.; Fotiadis, D.I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.* **1998**, *9*, 987–1000. [\[CrossRef\]](#)
29. Dissanayake, M.W.M.G.; Phan-Thien, N. Neural-network-based approximations for solving partial differential equations. *Commun. Numer. Methods Eng.* **1994**, *10*, 195–201. [\[CrossRef\]](#)
30. Rohrhofer, F.M.; Posch, S.; Geiger, B.C. On the Pareto Front of Physics-Informed Neural Networks. *arXiv* **2021**. [\[CrossRef\]](#)
31. Mohan, A.T.; Lubbers, N.; Livescu, D.; Chertkov, M. Embedding Hard Physical Constraints in Convolutional Neural Networks for 3D Turbulence. In Proceedings of the 8th International Conference on Learning Representations: Workshop on Tackling Climate Change with Machine Learning, ICLR'20, Addis Ababa, Ethiopia, 26 April–1 May 2020. [\[CrossRef\]](#)
32. Mattheakis, M.; Protopapas, P.; Sondak, D.; Di Giovanni, M.; Kaxiras, E. Physical Symmetries Embedded in Neural Networks. *arXiv* **2019**, arXiv:1904.08991.
33. Cuomo, S.; di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next. *J. Sci. Comput.* **2022**, *92*, 1–62. [\[CrossRef\]](#)
34. Lee, J.; Sohl-Dickstein, J.; Pennington, J.; Novak, R.; Schoenholz, S.; Bahri, Y. Deep Neural Networks as Gaussian Processes. In Proceedings of the 6th International Conference on Learning Representations, ICLR'18, Vancouver, BC, Canada, 30 April–3 May 2018. [\[CrossRef\]](#)

35. Jacot, A.; Gabriel, F.; Hongler, C. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Montreal, QC, Canada, 3–8 December 2018 ; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 8580–8589. [[CrossRef](#)]
36. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, ICML'16, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1050–1059. [[CrossRef](#)]
37. Novak, R.; Xiao, L.; Bahri, Y.; Lee, J.; Yang, G.; Hron, J.; Abolafia, D.A.; Pennington, J.; Sohl-Dickstein, J. Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes. In Proceedings of the 7th International Conference on Learning Representations, ICLR'19, New Orleans, LA, USA, 6–9 May 2019. [[CrossRef](#)]
38. Hron, J.; Bahri, Y.; Sohl-Dickstein, J.; Novak, R. Infinite attention: NNGP and NTK for deep attention networks. In Proceedings of the 37th International Conference on Machine Learning (ICML'20), Online, 12–18 July 2020; Volume 119, pp. 4376–4386. [[CrossRef](#)]
39. Hazan, T.; Jaakkola, T. Steps Toward Deep Kernel Methods from Infinite Neural Network. *arXiv* **2015**, arXiv:1508.05133.
40. Schaback, R.; Wendland, H. Kernel techniques: From machine learning to meshless methods. *Acta Numer.* **2006**, *15*, 543–639. [[CrossRef](#)]
41. Albert, C. Physics-Informed Transfer Path Analysis with Parameter Estimation using Gaussian Processes. In Proceedings of the 23rd International Congress on Acoustics, Aachen, Germany, 9–13 September 2019; Deutsche Gesellschaft für Akustik: Berlin, Germany, 2019; pp. 459–466. [[CrossRef](#)]
42. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR'15, San Diego, CA, USA, 7–9 May 2015. [[CrossRef](#)]