Periodic Activation Functions Induce Stationarity

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural network models are known to reinforce hidden data biases, making them unreliable and difficult to interpret. We seek to build models that 'know what they do not know' by introducing inductive biases in the function space. We show that *periodic* activation functions in Bayesian neural networks establish a connection between the prior on the network weights and translation-invariant, stationary Gaussian process priors. Furthermore, we show that this link goes beyond sinusoidal (Fourier) activations by also covering triangular wave and periodic ReLU activation functions. In a series of experiments, we show that periodic activation functions obtain comparable performance for in-domain data and capture sensitivity to perturbed inputs in deep neural networks for out-of-domain detection.

1 Introduction

Deep feedforward neural networks [43, 16] are an integral part of contemporary artificial intelligence and machine learning systems in visual and auditory perception, medicine, and general data analysis and decision making. However, when these methods have been adopted into real-world use, concerns related to robustness (with respect to data that has not been seen during training), fairness (hidden biases in data being reinforced by the model), and interpretability (why the model acts as it does) have taken a central role. The knowledge gathered by contemporary neural networks has even been characterised as never truly reliable [28]. These issues relate to the sensitivity of the trained model to perturbed inputs being fed through it—or the lack thereof.

This motivates *Bayesian deep learning*, where the interests are two-fold: encoding prior knowledge into models and performing probabilistic inference under the specified model. We focus on the former. Probabilistic approaches to specifying assumptions about the function space of deep neural networks have gained increasing attention in the machine learning community, compromising, among others, work analysing their posterior distribution [52, 1], discussing pathologies arising in uncertainty quantification [9], and calls for better Bayesian priors (*e.g.*, [39, 47, 34, 10]).

In this paper, we focus on stationary models, which act as a proxy for capturing *sensitivity*. Stationarity indicates translation-invariance, meaning that the joint probability distribution does not change when the inputs are shifted. This seemingly naive assumption has strong consequences in the sense that it induces *conservative* behaviour across the input domain, both in-distribution and outside the observed data. The resulting model is mean-reverting outside the training data (reversion to the prior), directly leading to higher uncertainty for out-of-distribution (OOD) samples (see Fig. 1 for examples). These features (together with some direct computational benefits) have made stationary models/priors the standard approach in kernel methods [4, 20], spatial statistics [5], and Gaussian process (GP) models [41], where the kernel is often chosen to induce stationarity.

Neural networks are parametric models, which typically fall into the class of non-stationary models. Non-stationarity increases *flexibility* and is often a sought-after property—especially if the interest is solely in optimizing for accuracy on in-domain test data. In fact, all standard neural network activation

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

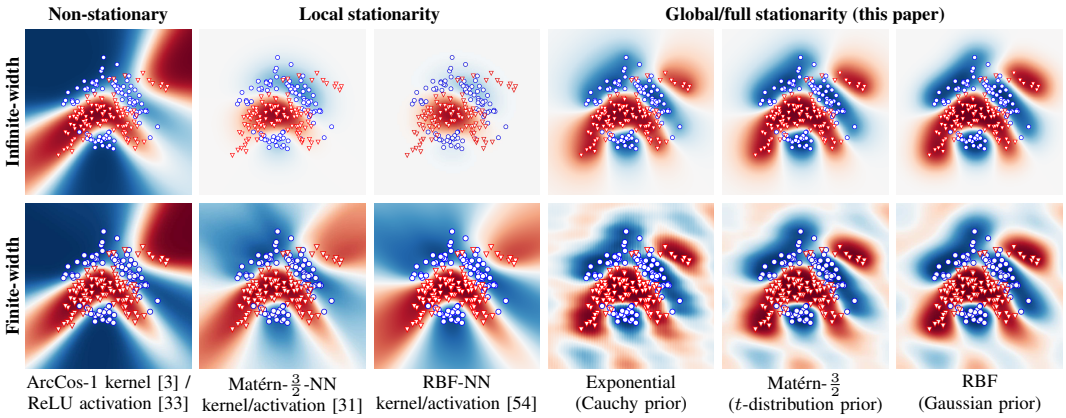


Figure 1: Posterior predictive densities of single hidden layer BNNs with 30 hidden units and their infinite-width corresponding GP on the `banana` classification task. Different activation functions induce different prior assumptions. Estimates obtained through HMC sampling [14] for 10k iterations.

functions (ReLU, sigmoid, *etc.*) induce non-stationarity (see [3, 54]). However, non-stationary models can easily induce over-confidence outside the observed data [18] or spurious relationships between input variables (as illustrated in Fig. 1, where darker shades show higher confidence).

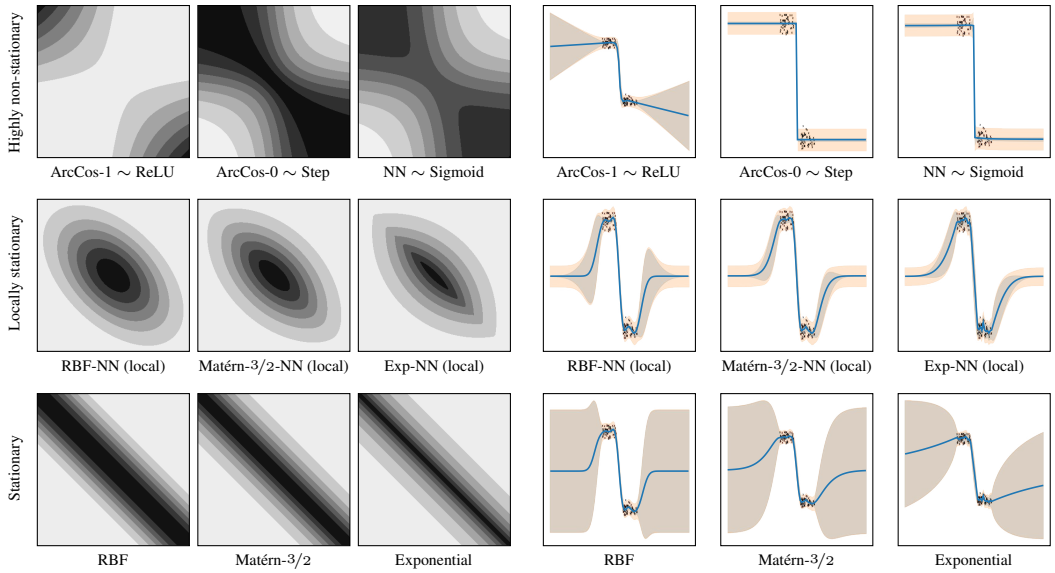
Stationary models in neural networks have been tackled before under the restriction of local stationarity, *i.e.*, translation-invariance is only *local*, induced by a Gaussian envelope (as realized by [54] for the RBF kernel/activation). Recently, Meronen et al. [31] expanded this approach and derived activation functions corresponding to the widely used Matérn class of kernels [29, 41]. We go one step further and derive activation functions that induce *global* stationarity. To do so, we leverage theory from harmonic analysis [53] of periodic functions, which helps expand the effective support over the entire input domain. We also realize direct links to previous works leveraging harmonic functions in neural networks [13, 2, 56, 44, 51], and Fourier features in other model families [40, 48].

The contributions of this paper are: *(i)* We show that periodic activation functions establish a direct correspondence between the prior on the network weights and the spectral density of the covariance function of the limiting stationary Gaussian process (GP) of single hidden layer Bayesian neural networks (BNNs). *(ii)* We leverage this correspondence and show that placing a Student-t prior on the weights of the hidden layer corresponds to a prior on the function space with Matérn covariance. *(iii)* Finally, we show in a range of experiments that periodic activation functions obtain comparable performance for in-domain data, do not result in overconfident predictions, and enable robust out-of-domain detection.

1.1 Related Work

We build upon prior work on exploring the covariance function induced by different activation functions, starting with the seminal work by Williams [54], who discussed a sigmoidal (ERF) and a Gaussian (RBF) activation function, resulting in a locally stationary kernel modulated by a Gaussian envelope. Cho and Saul [3] introduced non-stationary GP kernels corresponding to the ReLU and step activations, and [50] later extended the approach to the leaky ReLU and analysed alternative weight prior specifications. More recently, [31] derived activation functions corresponding to kernels from the Matérn family, which are locally stationary modulated Gaussian envelope.

At the same time, uninformative priors have been widely criticised in Bayesian deep learning [10], and alternative strategies have been proposed to incorporate prior knowledge. Pearce et al. [39] proposed compositions of BNNs to mimic compositional kernels in their limiting case. Sun et al. [47] proposed a variational objective for BNNs acting on the function space rather than the network parameters. Nalisnick et al. [34] proposed tractable priors on the functional complexity, obtained through the change of variables formula applied to the KL-divergence from a reference model/prior. And [32] proposed to encode prior assumptions on the activation level rather than the model parameters. However, none of the previous works explored settings under which the choice of activation function and prior induces a stationary prior in the function space.



(a) Gram matrices (b) Regression results with corresponding prior

Figure 2: **Left:** Gram/covariance matrices (for $\kappa(x, x')$) reflecting different prior beliefs on how infinite-width NN model functions ought to behave outside the observed data. **Right:** 1D regression results corresponding to the model induced by the priors, showing the posterior and predictive 95% intervals. See Fig. 7 in appendix for the corresponding finite-width results.

Recent works have shown the potential of periodic activation functions in neural networks, including approaches to learn implicit neural representation [44] and methods for recognition of handwritten digits [56]. Earlier work used periodic activation functions to resemble the Fourier series [13, 51] and showed that such models are universal function approximators [2]. In broader terms, [25] recently showed that the use of the Fourier transform, replacing self-attention, in Transformers can drastically improve the runtime. Inspired by the Fourier duality of covariance and spectral density, [40] proposed random Fourier features to approximate a GP with a RBF kernel using Monte Carlo integration. [48] later analysed the quality of the approximation, and various works have explored (random or deterministic) Fourier features, *e.g.*, in the context of kernel methods (*e.g.*, [23, 19, 45]).

To the best of our knowledge, none of the previous works have shown that periodic activation functions in BNNs establish a direct correspondence between the prior on the network weights and the density of the spectral density decomposition, thus, inducing global stationarity and enabling principled prior choices for BNNs.

2 Encoding Inductive Biases into Infinitely-wide Networks

We are concerned with the notions of *similarity* and *bias* in the input space when performing supervised learning. For this, we consider modelling assumptions present in random (untrained) models that induce the *a priori* (before observing data) characteristics in the model. We argue that these assumptions, when combined with data, either reinforce or suppress data biases depending on the prior assumptions. For analysis, we consider the mean function, $\mu(\cdot)$, and covariance (kernel) function, $\kappa(\cdot, \cdot)$, induced by model functions $f(\mathbf{x})$, representing the input–output mapping of a single-layer: $\mu(\mathbf{x}) \coloneqq \mathrm{E}[f(\mathbf{x})]$ and $\kappa(\mathbf{x}, \mathbf{x}') \coloneqq \mathrm{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}) - \mu(\mathbf{x}))^*]$. In probabilistic machine learning, rather than inferring the covariance structure from the expectations over model functions, one typically directly encodes assumptions by choosing a suitable parametric kernel family. However, both approaches, building a model by choosing a form for f or choosing a kernel family, implicitly or explicitly encode the same prior covariance structure.

Specification of prior knowledge through the covariance function and doing inference under this model, is at the core of Gaussian process (GP) models [41]. These models admit the form of a prior $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$ and a likelihood (observation) model $\mathbf{y} \mid \mathbf{f} \sim \prod_{i=1}^{n} p(y_i \mid f(\mathbf{x}_i))$, where the data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ are input–output pairs. This non-parametric machine learning

paradigm covers regression and classification tasks, under which GPs are known to offer a convenient machinery for learning and inference, while offering meaningful uncertainty estimates.

Neal [35] showed that an untrained single-layer network converges to a GP in the limit of infinite width. This link is well-understood and generalizes to deep models [7]. By placing an i.i.d. Gaussian prior on the weights and biases of a network, the distribution on the output of an untrained network converges to a Gaussian distribution in the limit of infinite layer width when applying the Central Limit Theorem. The joint distribution of outputs for any collection of inputs is multivariate Gaussian, as in a GP, and completely characterized by some kernel function $\kappa(\cdot,\cdot)$. Let $\sigma(\cdot)$ be some non-linear (activation) function, such as the ReLU or sigmoid, and $\mathbf{w}$ and b be the network weights and biases. Then, the associated kernel for the infinite-width network can be formulated in terms of [35, 12]:

$$\kappa(\mathbf{x},\mathbf{x}') = \int p(\mathbf{w})\, p(b)\, \sigma(\mathbf{w}^\mathsf{T}\mathbf{x} + b)\, \sigma(\mathbf{w}^\mathsf{T}\mathbf{x}' + b)\; \mathrm{d}\mathbf{w}\, \mathrm{d}b\,, \tag{1}$$

where priors $p(\mathbf{w})$ and $p(b)$ are chosen suitably. From the formulation in Eq. (1) one can read that the covariance function corresponding to an untrained (random) single-layer neural network is *fully characterized* by the choice of activation function $\sigma(\cdot)$ and the priors on the network weights.

Many of the covariance functions corresponding to commonly used activation functions under Gaussian priors on the weights have closed-form representations. The ArcCos kernel [3] covers the ReLU (ArcCos-1) and the step (ArcCos-0) activations, the so-called 'neural network kernel' [54, 41] links to sigmoidal activations. Gram matrices $\mathbf{K}_{ij} = \kappa(x_i, x'_j)$, evaluated for input pairs x and x' over $x, x' \in [-3, 3]$, are visualized in Fig. 2a to illustrate these covariance structures. The models induced by the ReLU, step, and sigmoid activations are by nature local, with the concentration of the non-linearity around the origin. As can be seen in the 1D regression results in Fig. 2b, the strong inductive bias in the prior (model) is reflected in the posterior. The ReLU extrapolates by matching the local bias of the data, while the step and sigmoidal saturate outside the data, and priors are clearly sensitive to translation and *highly non-stationary*.

Stationarity For a *stationary* (homogeneous) covariance function the covariance structure of the model functions $f(\mathbf{x})$ is the same regardless of the absolute position in the input space, and thus the parametrization can be relaxed to $\kappa(\mathbf{x},\mathbf{x}') \triangleq \kappa(\mathbf{x}-\mathbf{x}') = \kappa(\mathbf{r})$. Stationarity implies translation-invariance in the input space, *i.e.* the notion of similarity between inputs $\mathbf{x}$, and $\mathbf{x}'$ is only a function of their distance $\mathbf{x}-\mathbf{x}'$. This implies *weak stationarity* under stochastic process theory [37], and results in *reversion to the prior* outside observed data (see Fig. 2b). For stationary covariance functions, the best-known family is the Matérn [29, 46] family of kernels, which features models with sample functions of varying degrees of differentiability (smoothness):

$$\kappa_{\text{Mat.}}(\mathbf{x},\mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\, \frac{\|\mathbf{x}-\mathbf{x}'\|}{\ell}\right)^{\nu} \mathrm{K}_\nu\!\left(\sqrt{2\nu}\, \frac{\|\mathbf{x}-\mathbf{x}'\|}{\ell}\right), \tag{2}$$

where ν is a smoothness parameter, ℓ a characteristic length-scale parameter, $\mathrm{K}_\nu(\cdot)$ the modified Bessel function, and $\Gamma(\cdot)$ the gamma function. For the Matérn class, the processes are $\lceil \nu \rceil - 1$ times mean-square differentiable, and the family includes the RBF (squared exponential) and the exponential (Ornstein–Uhlenbeck) kernels as limiting cases as $\nu \to \infty$ and $\nu = {}^1\!/\!_2$.

Local stationarity Neural networks do not naturally exhibit stationarity. Formally, this stems from the problem of representing non-linearities over an infinite input domain with a finite set of local non-linear mappings. However, to bridge the gap between neural networks and widely-used stationary kernels, the typical approach is to restrict the invariance to be *local* (see Fig. 2a). This is the approach in the RBF-NN [55, 41] and general Matérn-NN [31] activation functions, where the prior is a composite covariance function with a Gaussian decay envelope (see discussion in [15]), *i.e.*,

$$\kappa_{\text{Mat-NN}}(\mathbf{x},\mathbf{x}') \propto \exp(-\mathbf{x}^\mathsf{T}\mathbf{x}/2\sigma_\mathrm{m}^2)\, \kappa_{\text{Mat.}}(\mathbf{x},\mathbf{x}')\, \exp(-\mathbf{x}'^\mathsf{T}\mathbf{x}'/2\sigma_\mathrm{m}^2)\,, \tag{3}$$

where $\sigma_\mathrm{m}^2 = 2\sigma_\mathrm{b}^2 + \ell^2$ (see Fig. 2a for the decay effect of the envelope). One motivation for this approach is to assume a Gaussian input density on the training/testing inputs, thus restricting our interest only to inputs in the vicinity of the training data. The local behaviour is highlighted in Fig. 2b, where the locally stationary models revert to the mean (as expected), but the marginal uncertainty of the model functions drops to zero when far enough from training samples. This motivates us to seek activation functions that induce (globally) stationary behaviour (bottom-rows in Fig. 2).

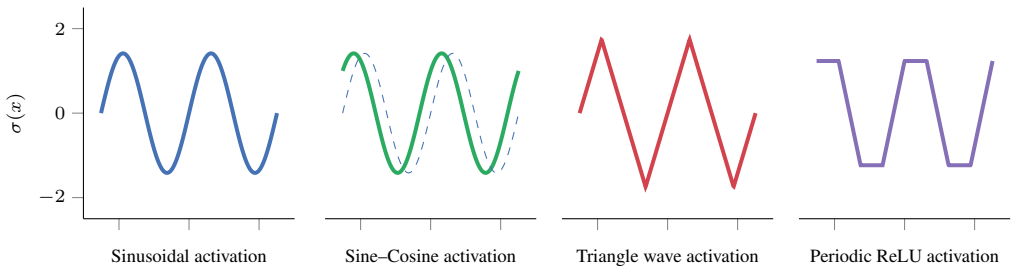


Figure 3: Sketches of types of periodic activations functions $\sigma(\cdot)$ visualized over two repeating periods each. The two leftmost are infinitely differentiable and the two rightmost are piecewise linear. The dashed line overlays the Sinusoidal activation for comparison.

3 Global Stationarity-inducing Activation Functions

We derive a direct connection between (fully) stationary covariance functions and periodic activation functions. Our main result leverages the spectral duality of the spectral density of stationary covariance functions, to establish a direct connection between the weight prior and the induced prior on the function space. Furthermore, we show that this connection is not restricted to the (sinusoidal) Fourier basis and establishes an equivalence relationship between a Student-t distribution on the weights and a prior on the function space with Matérn covariance structure.

3.1 Spectral Duality Under Harmonic Functions

For stationarity-inducing priors, the covariance function can be written equivalently in terms of its spectral density function. This stems from *Bochner's theorem* (see [6] and 36A in [26] for a more general formulation) which states a bounded continuous positive definite function $\kappa(\mathbf{r})$ can be represented through the following norm-preserving isomorphism

$$\kappa(\mathbf{r}) = \int \omega(\mathbf{r})\, \mu(\mathrm{d}\boldsymbol{\omega}), \tag{4}$$

where μ is a positive measure and $\mathbf{r} \in \mathbb{R}^d$. If the measure $\mu(\boldsymbol{\omega})$ has a density, it is called the *spectral density* $S(\boldsymbol{\omega})$ corresponding to the covariance function $\kappa(\mathbf{r})$. With $\omega(\mathbf{r}) = (2\pi)^{-d} \exp(\mathrm{i}\, \boldsymbol{\omega}^\mathsf{T}\mathbf{r})$, this gives rise to the Fourier duality of covariance and spectral density, which is known as the *Wiener–Khinchin theorem* (*e.g.*, [41]):

$$\kappa(\mathbf{r}) = \frac{1}{(2\pi)^d} \int S(\boldsymbol{\omega})\, e^{\mathrm{i}\, \boldsymbol{\omega}^\mathsf{T}\mathbf{r}}\, \mathrm{d}\boldsymbol{\omega} \quad \text{and} \quad S(\boldsymbol{\omega}) = \int \kappa(\mathbf{r})\, e^{-\mathrm{i}\, \boldsymbol{\omega}^\mathsf{T}\mathbf{r}}\, \mathrm{d}\mathbf{r}. \tag{5}$$

For $d > 1$, if the covariance function is *isotropic* (it only depends on the Euclidean norm $\|\mathbf{r}\|$ such that $\kappa(r) \triangleq \kappa(\|\mathbf{r}\|)$), then it is invariant to all rigid motions of the input space. Moreover, the spectral density will also only depend on the norm of the dual input variable $\boldsymbol{\omega}$. In the neural network case, we can assume the previous layer will take care of scaling the inputs, and thus we effectively are interested in the isotropic case, which brings us to analysing 1D projections.

An important question is whether we are restricted to the (sinusoidal) Fourier basis. Let $\psi_j(x)$ be any sequence of bounded, square-integrable (over its period) periodic functions, such that these functions define an inner product space (generalising a basis of a vector space). In signal processing terms, they define a *frame* that provides a redundant yet stable way of representing a signal. From a generalised harmonic perspective [53], we could resort to any convenient basis, such as Gabor wavelets [11]. However, in this work we seek to choose $\psi(x)$ suitably, such that we have uniform convergence to $\omega(r)$ (*cf.*, Eq. (4)), ensuring that we retain the spectral density $S(\omega)$.

3.2 Types of Periodic Activation Functions

We show how various periodic activations functions (see Fig. 4) under the view of Eq. (1) directly link back to the spectral duality. These periodic activation functions are continuous and bounded, while not necessarily differentiable and never monotonic over $\mathbb{R}$ (but monotonic within half-period).

Sinusoidal activation By assuming the activation function to be a sinusoid, $\sigma_{\sin}(x) = \sqrt{2}\sin(x)$ (see Fig. 4), placing a uniform prior on the bias term, $p(b) = \mathrm{Uniform}(-\pi, \pi)$, and substituting this into Eq. (1), we can obtain an expression for the covariance function $\kappa(x, x')$:

$$\kappa(x, x') = \int p(w) \int_{-\pi}^{\pi} \frac{1}{\pi} \sin(wx + b) \sin(wx' + b) \, \mathrm{d}b \, \mathrm{d}w \,. \tag{6}$$

By solving the inner integral and applying Euler's formula ($\cos(z) = \frac{1}{2}(e^{\mathrm{i}z} + e^{-\mathrm{i}z})$), we obtain:

$$\kappa(x, x') = \frac{1}{2} \int p(w) e^{\mathrm{i}w(x-x')} \, \mathrm{d}w + \frac{1}{2} \int p(w) e^{-\mathrm{i}w(x-x')} \, \mathrm{d}w \,, \tag{7}$$

which simplifies under the assumption that the prior on the weight is symmetric and has support on the real line, *i.e.*,

$$\kappa(x, x') = \int p(w) e^{\mathrm{i}w(x-x')} \, \mathrm{d}w \,. \tag{8}$$

By letting $r = x - x'$, we find that we recover the spectral density decomposition of a stationary process given by the Wiener–Khinchin theorem in Eq. (5), where $p(w) = \frac{1}{2\pi} S(w)$. Therefore, we obtain a direct connection between the prior on the weights and the spectral density. A detailed derivation can be found in App. A.1.

Sine–Cosine activation An alternative way of using sinusoidal activation functions is the use of a composite sine-cosine activation, *i.e.*, $\sigma_{\sin\cos}(x) = \sin(x) + \cos(x)$ (Fig. 4 compares the sin-cos to the sinusoidal activation). Using such construction has the benefit of removing the need to integrate over the bias, which can result in reduced variance [48] of the estimates, giving us a covariance function in the form of:

$$\begin{aligned} \kappa(x, x') &= \int p(w) \; [\sin(wx) + \cos(wx)] \, [\sin(wx') + \cos(wx')] \; \mathrm{d}w \\ &= \int p(w) \; \sin(w(x + x')) \, \mathrm{d}w + \int p(w) \; \cos(w(x - x')) \, \mathrm{d}w \,. \end{aligned} \tag{9}$$

By application of Euler's formula and under the usual assumption that $p(w)$ is symmetric and has support on the entire real line, the above reduces to:

$$\kappa(x, x') = \frac{1}{2} \left[\int p(w) \, e^{\mathrm{i}w(x-x')} \, \mathrm{d}w + \int p(w) \, e^{-\mathrm{i}w(x-x')} \, \mathrm{d}w \right] = \int p(w) \, e^{\mathrm{i}w(x-x')} \, \mathrm{d}w \;. \tag{10}$$

Again, by letting $r = x - x'$, we find that we recover the spectral density decomposition of a stationary process given by the Wiener–Khinchin theorem. A detailed derivation can be found in App. A.2.

Triangle wave activation Inspired by the success of piecewise linear activation functions (*e.g.*, ReLU [33], leaky ReLU [50], PReLU [17]), we show that the triangle wave, a periodic piecewise linear function, can be used instead of sinusoidal activations. The triangle wave is given by:

$$\psi(x) = \frac{4}{p} \left(x - \frac{p}{2} \left\lfloor \frac{2x}{p} + \frac{1}{2} \right\rfloor \right) (-1)^{\lfloor \frac{2x}{p} + \frac{1}{2} \rfloor} \,, \tag{11}$$

where p is the period and $\lfloor \bullet \rfloor$ denotes the floor function. By considering a period of $p = 2\pi$, assuming a uniform prior on the biases, $b \sim \mathrm{Uniform}(-\pi, \pi)$, and appropriate rescaling, we get the activation

$$\sigma_{\text{triangle}}(x) = \frac{\pi}{2\sqrt{2}} \left(x - \pi \left\lfloor \frac{x}{\pi} + \frac{1}{2} \right\rfloor \right) (-1)^{\lfloor \frac{x}{\pi} + \frac{1}{2} \rfloor} \,, \tag{12}$$

for which we obtain a direct correspondence of the network weight priors to the spectral density. As shown in the derivations in App. A.3, we find that we again recover the spectral density decomposition of a stationary process given by the Wiener–Khinchin theorem.

Periodic ReLU activation To go even further, we define a piecewise linear periodic activation function with a repeating rectified linear pattern, which we call the periodic ReLU function. It can be defined in terms of the sum of two triangle waves, with the second one being shifted by ¼ of a period. The resulting periodic function is piecewise linear and defined as:

$$\psi(x) = \frac{8}{\pi^2} \left(\left((x + \frac{\pi}{2}) - \pi \left\lfloor \frac{(x + \frac{\pi}{2})}{\pi} + \frac{1}{2} \right\rfloor \right) (-1)^{\lfloor \frac{(x + \frac{\pi}{2})}{\pi} + \frac{1}{2} \rfloor} + \left(x - \pi \left\lfloor \frac{x}{\pi} + \frac{1}{2} \right\rfloor \right) (-1)^{\lfloor \frac{x}{\pi} + \frac{1}{2} \rfloor} \right) , \tag{13}$$

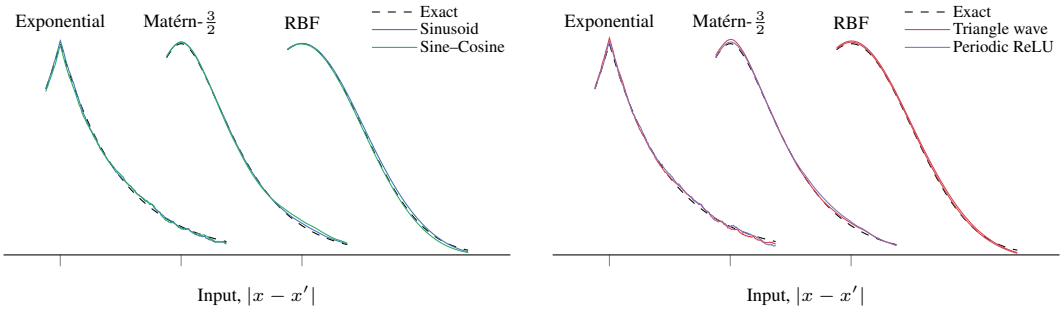


Figure 4: Covariance functions calculated by MC integration with 5000 samples and compared to their exact dashed counterparts (peaks shifted for clarity). Regardless of the type of periodic activation function used, we can recover the behaviour of the stationary kernels (see also Fig. 7a).

assuming a period of $p = 2\pi$. It is again possible to obtain a correspondence between the network weight prior and the spectral density by considering a weighted version of the equation above. In particular, we show in App. A.4 that from the following weighted periodic ReLU activation function,

$$\sigma_{\text{pReLU}}(x) = \frac{\pi}{4}\left(\left((x+\frac{\pi}{2})-\pi\left\lfloor\frac{(x+\frac{\pi}{2})}{\pi}+\frac{1}{2}\right\rfloor\right)(-1)^{\lfloor\frac{(x+\frac{\pi}{2})}{\pi}+\frac{1}{2}\rfloor}+\left(x-\pi\left\lfloor\frac{x}{\pi}+\frac{1}{2}\right\rfloor\right)(-1)^{\lfloor\frac{x}{\pi}+\frac{1}{2}\rfloor}\right), \tag{14}$$

we again recover the spectral density decomposition of a stationary process given by the Wiener–Khinchin theorem, providing a direct connection between the prior on the weights and the spectral density. Note that, choosing a piecewise linear periodic activation function has potential computational benefits compared to sinusoidal activation functions and can help prevent vanishing gradients.

3.3 Kernel Functions

We have established that is it possible to obtain a direct correspondence of the prior on the weights and the spectral density of a stationary covariance function by using periodic activation functions in random neural networks. In App. B we show that by placing a Student-t distribution on the weights with degrees of freedom of $u = 2\nu$ we recover the spectral density of the Matérn family, *i.e.*,

$$p(w) = \frac{\Gamma(\frac{u+1}{2})}{\sqrt{u\pi}\Gamma(\frac{u}{2})}\left(1+\frac{w^2}{u}\right)^{-\frac{u+1}{2}} = \frac{1}{2\pi}2\sqrt{\pi}\frac{\Gamma(\nu+\frac{1}{2})}{\Gamma(\nu)}(2\nu)^{\nu}\left(2\nu+w^2\right)^{-(\nu+\frac{1}{2})} = \frac{1}{2\pi}S_{\text{Mat.}}(w)\,, \tag{15}$$

where $p(w)$ denotes the probability density function a Student-t distribution and $S_{\text{Mat.}}(w)$ denotes the spectral density of the Matérn family. This means that a Student-t prior on the weights in combination with an appropriately scaled periodic activation function corresponds directly to a prior in the function space with Matérn covariance structure. Fig. 4 verifies this result and shows that we recover the exact covariance function (dashed) for various examples from the Matérn family by Monte Carlo (MC) integration (5000 samples) with all of the discussed periodic activation functions.

Recall that the Matérn family contains the exponential kernel ($\nu = 1/2$) and the RBF Kernel ($\nu = \infty$) as special cases. Similarly, the Student-t distribution has the Cauchy ($u = 1$) and the Normal distribution ($u = \infty$) as limiting cases, resulting in a correspondence to the respective special cases in the Matérn family. Table 2 in App. B summarises the set of priors on the weights that correspond to the spectral density of kernel functions in the Matérn family.

4 Experiments

To assess the performance of stationarity-inducing activation functions compared to non-stationary and locally stationary models, we quantified the in-domain and out-of-domain predictive uncertainty on various benchmark data sets. In all experiments, we compare each of the different models by only replacing the activation function and the respective priors of the last hidden layer of the network. A detailed description of the experimental setup is available in App. D. The illustrative toy BNN examples are implemented using HMC in Turing.jl [14], GP regression results use GPflow [30], and all other experiments are implemented using PyTorch [38].

Table 1: Examples of UCI regression tasks, showing the globally stationary NN model directly gives competitive RMSE and for most data sets also improved mean negative log predictive density (NLPD) compared to non-stationary and locally stationary NN models. KFAC Laplace was used as the inference method. High std is due to variability in 10-fold-cv splits, not method performance.

(n, d)	BOSTON (506, 12)		CONCRETE (1030, 5)		AIRFOIL (1503, 5)		ELEVATORS (16599, 18)	
(c, n_{batch})	(1, 50)		(1, 50)		(1, 50)		(1, 500)	
	NLPD	RMSE	NLPD	RMSE	NLPD	RMSE	NLPD	RMSE
ReLU	0.51 ± 0.32	0.37 ± 0.07	0.78 ± 0.16	0.48 ± 0.04	0.51 ± 0.53	0.41 ± 0.21	0.38 ± 0.03	0.35 ± 0.01
loc RBF	0.52 ± 0.30	0.37 ± 0.08	0.78 ± 0.22	0.44 ± 0.05	0.10 ± 0.15	0.26 ± 0.03	0.41 ± 0.04	$\mathbf{0.35\pm0.01}$
glob RBF (sin)	0.42 ± 0.34	0.36 ± 0.07	0.74 ± 0.15	0.49 ± 0.05	0.14 ± 0.17	0.29 ± 0.05	0.38 ± 0.03	0.35 ± 0.01
glob RBF (prelu)	0.39 ± 0.30	$\mathbf{0.36\pm0.07}$	0.74 ± 0.14	0.49 ± 0.04	$\mathbf{0.05\pm0.12}$	$\mathbf{0.26\pm0.03}$	0.74 ± 0.73	0.46 ± 0.21
loc Mat-3/2	0.71 ± 0.38	0.40 ± 0.08	0.84 ± 0.28	$\mathbf{0.42\pm0.04}$	0.11 ± 0.18	0.26 ± 0.03	0.43 ± 0.04	0.35 ± 0.01
glob Mat-3/2 (sin)	0.43 ± 0.27	0.39 ± 0.08	0.73 ± 0.16	0.49 ± 0.05	0.07 ± 0.15	0.27 ± 0.03	$\mathbf{0.37\pm0.03}$	0.35 ± 0.01
glob Mat-3/2 (prelu)	$\mathbf{0.38\pm0.22}$	0.38 ± 0.08	$\mathbf{0.72\pm0.18}$	0.48 ± 0.05	0.08 ± 0.12	0.27 ± 0.03	0.39 ± 0.03	0.36 ± 0.01

Illustrative toy examples Fig. 1 shows predictive densities for non-stationary, locally stationary, and globally stationary activation functions on the banana classification task. The top row illustrates the predictive densities of infinite-width BNNs (GP), the bottom row shows corresponding results for a single hidden layer BNN using 30 hidden units. We observe that global stationarity-inducing activation functions revert to the prior outside the data, leading to conservative behaviour (high uncertainty) for out-of-domain samples. Moreover, we see that the finite-width BNNs result in similar behaviour to their infinite-width counterpart, while locally stationary activation functions in finite-width BNNs exhibit a slower reversion to the mean than their infinite-width corresponding GPs. Additionally, we include a 1D toy regression study that highlights the differences between different prior assumptions encoded by choice of the activation function. Fig. 2 shows the corresponding prior covariance as well as posterior predictions for the infinite-width (GP) model. We replicated the exact same study with a finite-width network and recover the same behaviour (see Fig. 7 in App. D.1).

UCI benchmarks Table 1 shows results on UCI [8] regression data sets comparing deep neural networks with ReLU, locally stationary RBF [55], and locally stationary Matérn-$^3/_2$ [31] against global stationary models. An extended results table can be found in App. D.2 and additional results on UCI classification tasks in App. D.3. Table 1 lists root mean square error (RMSE) and negative log predictive density (NLPD), which captures the predictive uncertainty, while the RMSE only accounts for the mean. Global stationary models provide better estimates of the target distribution in all cases while obtaining comparable RMSEs. Posterior inference is performed using KFAC Laplace [42].

Detection of distribution shift with rotated MNIST Fig. 5 demonstrates the predictive behaviour of different activation functions under covariate shift. We evaluate the predictive accuracy, mean confidence of the predicted class, and NLPD on MNIST [24] for different rotations of the input. Models are trained only on unrotated digits using KFAC Laplace [42]. The results indicate that all models obtain similar accuracy results, while only local and global stationary models do not result in over-confident uncertainty estimates. For an ideally calibrated model, the mean confidence would decrease as low as the accuracy curve when the digits are rotated, which would keep the NLPD values as low as possible.

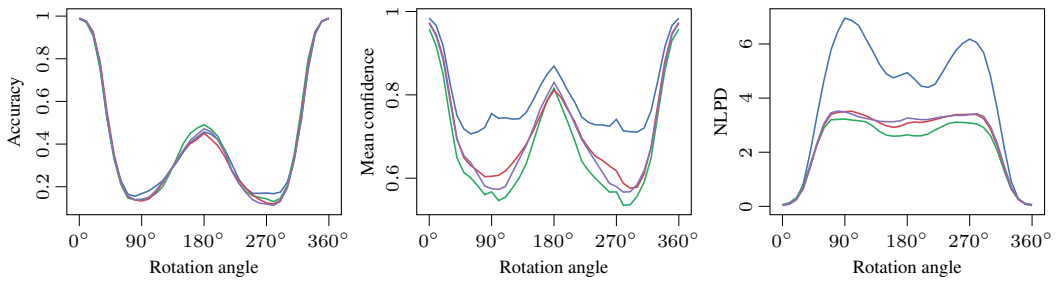


Figure 5: Rotated MNIST: The models have been trained on unrotated digits. The test-set digits are rotated at test time to show the sensitivity of the trained model to perturbations. All models perform equally in terms of accuracy, while ReLU (——) shows overconfidence in terms of mean confidence and NLPD. The stationary RBF models (—— local, —— sin, —— sin-cos) capture uncertainty.

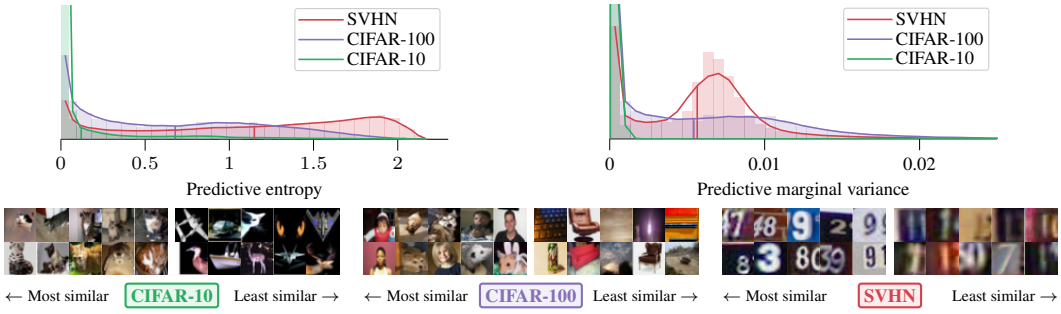


Figure 6: Good OOD separation with a globally stationary Matérn-$^3/_2$ model trained on CIFAR-10 and tested on CIFAR-10, CIFAR-100, and SVHN. Predictive entropy histograms of test image results are on the left, and predictive marginal variance histograms are on the right. On the bottom are sample images from each test set: left-side images with lowest entropy/highest confidence, and right-side images with highest entropy/lowest confidence. See App. D.5 for histograms for other models.

Out-of-distribution detection using CIFAR-10, CIFAR-100, and SVHN Fig. 6 shows globally stationary Matérn-3/2 model performance on out-of-distribution detection for image classification. The model has been trained using SWAG [27] on CIFAR-10 [22], and tested on CIFAR-10, CIFAR-100 [22], and SVHN [36] test set images. Both CIFAR-100 (more similar) and SVHN (more dissimilar to CIFAR-10) images are OOD data, and the model should show high uncertainties for the respective test images. The histograms of predictive entropies for different test sets show that the globally stationary model can separate between in-distribution and OOD data. The predictive marginal variance histograms show that the model has higher variance on OOD data compared to in-distribution data. Both entropy and variance histograms show that the model considers SVHN more OOD than CIFAR-100, which is intuitive as CIFAR-100 resembles CIFAR-10 more. Moreover, we found that overall the globally stationary models work better on the OOD detection task, compared to locally stationary and ReLU models, as expected (see Fig. 10 in App. D.5 for model comparison).

5 Discussion and Conclusions

We have shown that *periodic* activation functions in Bayesian neural networks (BNNs) establish a direct connection between the prior on the network weights and the spectral density of the limiting stationary process. This link goes beyond the Fourier (sinusoidal) basis, which we have illustrated by deriving the correspondence also for the triangle wave and a new periodic variant of the ReLU activation function. Moreover, we have shown that for BNNs with a periodic activation function, placing a Student-t distribution on the network weights corresponds to a prior in the function space with Matérn covariance structure.

Our work can help build neural network models that are more robust and less vulnerable to OOD data—occurring either naturally or maliciously. As demonstrated in the experiments, the modelling assumptions have practical importance in reducing sensitivity to input shift. Perturbation of inputs can be captured in the last layer of a deep model, and the principled choice of *a priori* stationarity has shown to be effective based on the experiments. Using periodic activation functions in BNNs induces *global* stationarity, *i.e.*, the model is not only locally translation-invariant, which induces conservative behaviour across the input domain. The resulting model reverts to the prior for out-of-distribution (OOD) samples, resulting in well-specified uncertainties.

In contrast to models with non-stationary or locally stationary activations, optimising and performing approximate inference in models with periodic activation functions has proven to be more challenging. In particular, we found that the uncertainty over the bias terms is often not accurately estimated when using KFAC Laplace or SWAG, which required us to use a relatively large number of hidden units. However, the experiments using dynamic HMC indicate that the number of hidden units can be drastically reduced using a more accurate approximate inference scheme.

Upon acceptance, the codes and data to replicate the results will be available under MIT license at `http://github.com/anonymized` (see supplement for now).

References

[1] L. Aitchison. A statistical theory of cold posteriors in deep neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[2] E. J. Candès. Harmonic analysis of neural networks. *Applied and Computational Harmonic Analysis*, 6(2): 197–218, 1999.

[3] Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 342–350. Curran Associates, Inc., 2009.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[5] N. A. Cressie. *Statistics for Spatial Data*. Wiley series in probability and mathematical statistics. Wiley-Interscience, 1991.

[6] G. Da Prato and J. Zabczyk. *Stochastic Equations in Infinite Dimensions*, volume 45 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1992.

[7] A. G. de G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[8] D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[9] A. Y. Foong, D. R. Burt, Y. Li, and R. E. Turner. On the expressiveness of approximate inference in Bayesian neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 15897–15908. Curran Associates, Inc., 2020.

[10] V. Fortuin, A. Garriga-Alonso, F. Wenzel, G. Rätsch, R. E. Turner, M. van der Wilk, and L. Aitchison. Bayesian neural network priors revisited. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021.

[11] D. Gabor. Theory of communication part 1: The analysis of information. *Journal of the Institution of Electrical Engineers–Part III: Radio and Communication Engineering*, 93(26):429–441, 1946.

[12] Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059. PMLR, 2016.

[13] A. R. Gallant and H. White. There exists a neural network that does not make avoidable mistakes. In *Proceedings of International Conference on Neural Networks (ICNN)*, pages 657–664, 1988.

[14] H. Ge, K. Xu, and Z. Ghahramani. Turing: A language for flexible probabilistic inference. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84 of *Proceedings of Machine Learning Research*, pages 1682–1690. PMLR, 2018.

[15] M. G. Genton. Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2001.

[16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[18] M. Hein, M. Andriushchenko, and J. Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 41–50, 2019.

[19] J. Hensman, N. Durrande, and A. Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18:1–52, 2018.

[20] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, pages 1171–1220, 2008.

[21] A. Kristiadi, M. Hein, and P. Hennig. Being bayesian, even just a bit, fixes overconfidence in ReLU networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. PMLR, 2020.

[22] A. Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2012.

[23] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

[24] Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits, 1998. URL `http://yann.lecun.com/exdb/mnist/`.

[25] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontañón. Fnet: Mixing tokens with Fourier transforms. *ArXiv preprint arXiv:1401.5508*, 2021.

[26] L. Loomis. *An Introduction to Abstract Harmonic Analysis*. D. van Nostrand Company, Inc., New York, 1954.

[27] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 13132–13143. Curran Associates, Inc., 2019.

[28] G. Marcus. The next decade in AI: Four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*, 2020.

[29] B. Matérn. Spatial variation: Stochastic models and their applications to some problems in forest surveys and other sampling investigations. *Meddelanden från statens skogsforskningsinstitut*, 49:1–144, 1960.

[30] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017.

[31] L. Meronen, C. Irwanto, and A. Solin. Stationary activations for uncertainty calibration in deep learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 2338–2350. Curran Associates, Inc., 2020.

[32] P. Morales-Alvarez, D. Hernández-Lobato, R. Molina, and J. M. Hernández-Lobato. Activation-level uncertainty in deep neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[33] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 807–814. Omnipress, 2010.

[34] E. T. Nalisnick, J. Gordon, and J. M. Hernández-Lobato. Predictive complexity priors. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130 of *Proceedings of Machine Learning Research*, pages 694–702. PMLR, 2021.

[35] R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, Toronto, Canada, 1995.

[36] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. 2011. URL `http://ufldl.stanford.edu/housenumbers`.

[37] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1991.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019.

[39] T. Pearce, R. Tsuchida, M. Zaki, A. Brintrup, and A. Neely. Expressive priors in Bayesian neural networks: Kernel combinations and periodic functions. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 115 of *Proceedings of Machine Learning Research*, pages 134–144. PMLR, 2019.

[40] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 1177–1184, 2008.

[41] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[42] H. Ritter, A. Botev, and D. Barber. A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[43] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[44] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 7462–7473. Curran Associates, Inc., 2020.

[45] A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.

[46] M. L. Stein. *Interpolation of Spatial Data*. Springer Series in Statistics. Springer, New York, 1999.

[47] S. Sun, G. Zhang, J. Shi, and R. B. Grosse. Functional variational Bayesian neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[48] D. J. Sutherland and J. G. Schneider. On the error of random Fourier features. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 862–871. AUAI Press, 2015.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[50] R. Tsuchida, F. Roosta, and M. Gallagher. Invariance of weight distributions in rectified MLPs. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 4995–5004. PMLR, 2018.

[51] M. Uteuliyeva, A. Zhumekenov, R. Takhanov, Z. Assylbekov, A. J. Castro, and O. Kabdolov. Fourier neural networks: A comparative study. *Intelligent Data Analysis*, 24(5):1107–1120, 2020.

[52] F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin. How good is the Bayes posterior in deep neural networks really? In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 10248–10259. PMLR, 2020.

[53] N. Wiener. Generalized harmonic analysis. *Acta Mathematica*, 55:117–258, 1930.

[54] C. K. I. Williams. Computing with infinite networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 295–301. MIT Press, 1997.

[55] C. K. I. Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

[56] K. Wong, C. Leung, and S. Chang. Handwritten digit recognition using multi-layer feedforward neural networks with periodic and monotonic activation functions. In *Proceedings of International Conference on Pattern Recognition (ICNN)*, pages 106–109, 2002.

[57] I.-C. Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28:1797–1808, 1998.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See main text and App. A
   (b) Did you include complete proofs of all theoretical results? [Yes] See App. A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 4 and App. D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Sec. 4 and App. D.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See App. D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Sec. 4 and App. D.
   (b) Did you mention the license of the assets? [Yes] See App. D.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include only a reference implementation, no new data assets are released.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]