Neural Networks as Inter-Domain Inducing Points

Shengyang Sun[*] SSY@CS.TORONTO.EDU
University of Toronto, Vector Institute

Jiaxin Shi[*] JIAXINSHI@MICROSOFT.COM
Microsoft Research New England

Roger Grosse RGROSSE@CS.TORONTO.EDU
University of Toronto, Vector Institute

Abstract

Equivalences between infinite neural networks and Gaussian processes have been established for explaining the functional prior and training dynamics of deep learning models. In this paper we cast the hidden units of finite-width neural networks as the inter-domain inducing points of a kernel, then a one-hidden-layer network becomes a kernel regression model. For dot-product kernels on both $\mathbb{R}^d$ and $\mathbb{S}^{d-1}$, we derive the kernel functions for inducing points. Empirically we conduct toy experiments to validate the proposed approaches.

1. Introduction

Connections between Gaussian processes (GP) and deep neural networks (DNN) have been drawn to explain the success of deep learning. Specifically, infinite-width neural networks have been demonstrated equivalent to Gaussian processes of a NNGP kernel (Neal, 1995; Lee et al., 2018), which state that the prior of a infinite random neural network is a GP prior over functions. Furthermore, neural tangent kernels (NTK) (Jacot et al., 2018; Arora et al., 2019) are shown to govern the training dynamics of infinite neural networks. Given these equivalences between infinite neural networks and Gaussian processes, finite-width neural networks can be regarded as random feature approximations (Rahimi and Recht, 2008; Ghorbani et al., 2020) of the corresponding GP. Using the Jacobians as feature maps, Khan et al. (2019) also associate finite Bayesian neural networks to equivalent Gaussian processes.

For a finite-width one-hidden-layer network, the theory of random feature approximations requires the weights of the first layer to be fixed after initialization, which is contrary to the real training and cannot learn adaptive representations of the data manifold. In this paper we cast an alternative perspective, formulating the one-hidden-layer networks as conducting Nystrom approximations (Drineas and Mahoney, 2005) instead of random feature approximations. Specifically, for a kernel k, each hidden unit $\sigma(\mathbf{w}_m^\top \mathbf{x})$ corresponds to an inter-domain inducing point z_m (Titsias, 2009; Lázaro-Gredilla and Figueiras-Vidal, 2009). The overall network $f(\mathbf{x}) = \sum_{m=1}^M \mathbf{a}_i \sigma(\mathbf{w}_m^\top \mathbf{x})$ is then equivalently $f(\mathbf{x}) = \sum_{m=1}^M \mathbf{a}_i k(\mathbf{x}, z_m)$. In such way, the fully connected network is formulated as a kernel regression model, where the weights $\{\mathbf{w}_m\}_{m=1}^M$ in the first layer are M inducing locations and the weights $\mathbf{a}$ in the second layer are linear coefficients.

[*] Equal contribution. Author ordering determined by coin flip.

2. Background: Inter-domain GP and Variational Fourier Features

Inter-domain Gaussian processes (Lázaro-Gredilla and Figueiras-Vidal, 2009) introduce the inducing points u_z by integrating with an inducing function $z(\mathbf{x})$,

$$u_z = \int f(\mathbf{x})z(\mathbf{x})d\mathbf{x}, \tag{1}$$

Depending on the inducing function $z(\mathbf{x})$, the inter-domain inducing points lead to various kernel influences $k_{\mathbf{x}}(\cdot)$, which can be obtained via,

$$k(z,\mathbf{x}) = \mathbb{E}_f[f(\mathbf{x})u_z] = \int k(\mathbf{x},\mathbf{x}')z(\mathbf{x}')d\mathbf{x}', \tag{2}$$

$$k(z,z') = \mathbb{E}_f[u_z u_{z'}] = \int k(\mathbf{x},\mathbf{x}')z(\mathbf{x})z'(\mathbf{x}')d\mathbf{x}d\mathbf{x}', \tag{3}$$

Evaluating the kernels $k(z,\mathbf{x}), k(z,z')$ requires integrations, which are usually intractable. Variational Fourier Features (VFF) (Hensman et al., 2017) propose a variation of the inter-domain GPs by directly specifying the kernel function $k(z,\mathbf{x})$. In concrete, let the inducing function $z \in \mathcal{H}$ belong to the reproducing kernel Hilbert space $\mathcal{H}$ of k, VFF defines the inducing points by the RKHS inner product instead of the ℓ^2 inner product,

$$u_z = \langle f, z\rangle_{\mathcal{H}}, \tag{4}$$

Since the RKHS inner product is also a linear operator, the kernel functions can be computed,

$$k(z,\mathbf{x}) = \mathbb{E}[u_z f(\mathbf{x})] = \mathbb{E}[\langle f * f(\mathbf{x}), z\rangle_{\mathcal{H}}] = \langle k(\mathbf{x},\cdot), z\rangle_{\mathcal{H}} = z(\mathbf{x}), \tag{5}$$

$$k(z,z') = \mathbb{E}[u_z u_{z'}] = \mathbb{E}[\langle f, z\rangle_{\mathcal{H}}\langle f, z'\rangle_{\mathcal{H}}] = \langle z, z'\rangle_{\mathcal{H}}. \tag{6}$$

The kernel function $k(z,\mathbf{x})$ is exactly the inducing function z, and $k(z,z')$ is the RKHS inner product between z and z'. Therefore, how to choose the inducing functions $z_1, ..., z_M$ affects the accuracy of variational approximations and the computational cost. Specifically, Hensman et al. (2017) applies sinusoidal Fourier features on a one-dimensional bounded segment $\mathcal{H}_{[a,b]}$. Furthermore, Burt et al. (2020) propose orthogonal features for stationary kernels $k(\mathbf{x},\mathbf{x}) = \kappa(\mathbf{x}-\mathbf{x}')$, whose resulting kernel matrix $\mathbf{K_{zz}}$ is diagonal. Dutordoir et al. (2020) propose to use the spherical harmonics as inducing functions for zonal kernels $k(\mathbf{x},\mathbf{x}) = \kappa(\mathbf{x}^\top\mathbf{x}')$. Because the spherical harmonics are eigenfunctions of zonal kernels, they are also orthogonal to each other. However, the aforementioned variational features are all data-independent, which might suffer the curse-of-dimensionality issue.

3. Neural Networks as Inter-domain Inducing Points

In this section we adopt the theory of variational Fourier features and reformulate neural networks as inter-domain inducing points. Specifically, we can explain each neuron $\sigma(\mathbf{w}^\top\mathbf{x})$ as one inducing function z in VFF. Compared to previous variational features (Hensman et al., 2017; Burt et al., 2020; Dutordoir et al., 2020), our proposed features are fully trainable and naturally adapt to high dimensions. Specifically, for d-dimensional inputs, a one-hidden-layer neural network is defined as the function,

$$f(\mathbf{x}) = \sum_{m=1}^{M} \mathbf{a}_m \sigma(\mathbf{w}_m^\top \mathbf{x}), \tag{7}$$

where σ is the activation function and M is the number of hidden units. $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_M] \in \mathbb{R}^{d\times M}$ and $\mathbf{a} \in \mathbb{R}^M$ are trainable parameters. We assume a dot-product kernel $k(\mathbf{x}, \mathbf{x}') = \tau(\mathbf{x}^\top \mathbf{x}')$, where τ is a positive-type function (Berlinet and Thomas-Agnan, 2011). For $\mathbf{w} \in \mathbb{R}^d$, we define the inducing function z as $\sigma(\mathbf{w}^\top \mathbf{x})$. Then the network has M inducing functions.

3.1. Dot-Product Kernels on $\mathbb{R}^d$

In this subsection we assume the input domain $\mathcal{X} = \mathbb{R}^d$. As shown by Smola et al. (2001), the analytic kernel k is positive semidefinite if and only if τ admits a Taylor expansion with nonnegative coefficients, i.e., $\tau(t) = \sum_{j=0}^{\infty} \beta_j t^j, \beta_j \geq 0$. We further assume the Taylor expansion of the activation, $\sigma(t) = \sum_{j=0}^{\infty} \alpha_j t^j$. Given the Taylor expansions, the kernel admits the feature map,

$$k(\mathbf{x}, \mathbf{x}') = \sum_{j=0}^{\infty} \beta_j (\mathbf{x}^\top \mathbf{x}')^j = \sum_{j=0}^{\infty} \beta_j \sum_{1\leq i_1,...,i_j\leq d} \mathbf{x}_{i_1} \cdots \mathbf{x}_{i_j} \mathbf{x}'_{i_1} \cdots \mathbf{x}'_{i_j} = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle, \tag{8}$$

Where the feature map $\varphi(\mathbf{x}) = [\sqrt{\beta}_j \mathbf{x}_{i_1} \cdots \mathbf{x}_{i_j}]_{j,i_1,...,i_j}$. The variational feature $\sigma(\mathbf{w}^\top \mathbf{x})$ can be represented as an inner product with the feature map φ. Specifically,

$$\sigma(\mathbf{w}^\top \mathbf{x}) = \sum_{j=0}^{\infty} \alpha_j (\mathbf{w}^\top \mathbf{x})^j = \sum_{j=0}^{\infty} \alpha_j \sum_{1\leq i_1,...,i_j\leq d} \mathbf{x}_{i_1} \cdots \mathbf{x}_{i_j} \mathbf{w}_{i_1} \cdots \mathbf{w}_{i_j} \tag{9}$$

$$= \sum_{j=0}^{\infty} \sum_{1\leq i_1,...,i_j\leq d} \left(\frac{\alpha_j}{\sqrt{\beta_j}} \mathbf{w}_{i_1} \cdots \mathbf{w}_{i_j} \right) \left(\sqrt{\beta_j} \mathbf{x}_{i_1} \cdots \mathbf{x}_{i_j} \right) = \langle \zeta(\mathbf{w}), \varphi(\mathbf{x}) \rangle. \tag{10}$$

where the weights $\zeta(\mathbf{w}) = [\frac{\alpha_j}{\sqrt{\beta_j}} \mathbf{w}_{i_1} \cdots \mathbf{w}_{i_j}]_{j,i_1,...,i_j}$. Because each term of the feature map $\varphi(\mathbf{x})$ is one independent monomial, the RKHS inner product between $\sigma(\mathbf{w}^\top \mathbf{x})$ and $\sigma(\mathbf{w}'^\top \mathbf{x})$ equals to the ℓ^2 inner product of $\zeta(\mathbf{w})$ and $\zeta(\mathbf{w}')$.

$$\langle \sigma(\mathbf{w}^\top \mathbf{x}), \sigma((\mathbf{w}')^\top \mathbf{x}) \rangle_{\mathcal{H}} = \langle \zeta(\mathbf{w}), \zeta(\mathbf{w}') \rangle = \sum_{j=0}^{\infty} \frac{\alpha_j^2}{\beta_j} \sum_{1\leq i_1,...,i_j\leq d} \mathbf{w}'_{i_1} \cdots \mathbf{w}'_{i_j} \mathbf{w}_{i_1} \cdots \mathbf{w}_{i_j} \tag{11}$$

$$= \sum_{j=0}^{\infty} \frac{\alpha_j^2}{\beta_j} \left(\mathbf{w}^\top \mathbf{w}' \right)^j = \rho(\mathbf{w}^\top \mathbf{w}'). \tag{12}$$

where we define $\rho(t) = \sum_{j=0}^{\infty} \frac{\alpha_j^2}{\beta_j} t^j$. Therefore, the kernel function $k(\sigma(\mathbf{w}^\top \cdot), \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x})$ and $k(\sigma(\mathbf{w}^\top \cdot), \sigma(\mathbf{w}'^\top \cdot)) = \rho(\mathbf{w}^\top \mathbf{w}')$.

3.2. Dot-Product Kernels on $\mathbb{S}^{d-1}$

In this subsection we assume the input lies on the sphere $\mathbb{S}^{d-1} = \{\mathbf{x} | \mathbf{x} \in \mathbb{R}^d, \|\mathbf{x}\|_2 = 1\}$. Compared to the previous subsection, the change of input domain has substantial influences. For example, the neural network kernel (NNGP) (Neal, 1995; Lee et al., 2018) and the neural tangent kernel (NTK) (Jacot et al., 2018) depend on the norms $\mathbf{x}^\top \mathbf{x}, \mathbf{x}'^\top \mathbf{x}$ on $\mathbb{R}^d$, thus the previous analyses do not apply. Furthermore, if we restrict $\mathbf{x} \in \mathbb{S}^{d-1}$, the previous feature map terms $\sqrt{\beta}_j \mathbf{x}_{i_1} \cdots \mathbf{x}_{i_j}$ in $\varphi(\mathbf{x})$ are not independent monomials anymore, thus the derived RKHS inner products are not applicable either.

The theory of spherical harmonics (Thomson and Tait, 1888; Morimoto, 1998) is an important tool for analyzing functions on sphere, which we will adopt for computing the RKHS inner products. Denote by $P_n^d(\xi)$ the associated Legendre polynomials on $[-1, 1]$, which form a complete orthogonal basis for functions on $\mathbb{S}^{d-1}$. The Addition formula (see e.g. Gallier, 2009, Proposition 1.18) states that, if $\{Y_{n,j}\}_{n,j}$ are a set of spherical harmonics on $\mathbb{S}^{d-1}$, for any $\mathbf{x}, \mathbf{x}' \in \mathbb{S}^{d-1}$,

$$\text{The Addition Formula: } P_n^d(\mathbf{x} \cdot \mathbf{x}') = \frac{|\mathbb{S}^{d-1}|}{N(d,n)} \sum_{j=1}^{N(d,n)} Y_{n,j}(\mathbf{x}) Y_{n,j}(\mathbf{x}'). \tag{13}$$

Where $|\mathbb{S}^{d-1}|$ is the area of $\mathbb{S}^{d-1}$, and $N(d,n) = \binom{n+d-1}{n} - \binom{n+d-3}{n-2}$ is the number of spherical harmonics of degree n. Because $\{P_n^d(\xi)\}_{n=0}^\infty$ is a complete orthogonal basis, the Funk-Hecke formula (Smola et al., 2001; Gallier, 2009; Müller, 2012) states, for any analytic function f on $[-1, 1]$,

$$\text{The Funk-Hecke Formula: } f(\xi) = \sum_{n=0}^{\infty} N(d,n) \frac{|\mathbb{S}^{d-2}|}{|\mathbb{S}^{d-1}|} P_n^d(\xi) \hat{f}(n), \tag{14}$$

where $\hat{f}(n) = \int_{-1}^{1} f(\xi') P_n^d(\xi') (1 - \xi'^2)^{\frac{d-3}{2}} d\xi'$ [1]. Now we can use these established results in spherical harmonics for the proposed variational features. Since $k(\mathbf{x}, \mathbf{x}') = \tau(\mathbf{x}^\top \mathbf{x}')$, we apply the Funk-Hecke formula for τ and rewrite P_n^d using the Addition formula,

$$k(\mathbf{x}, \mathbf{x}') = \sum_{n=0}^{\infty} |\mathbb{S}^{d-2}| \sum_{j=1}^{N(d,n)} Y_{n,j}(\mathbf{x}) Y_{n,j}(\mathbf{x}') \hat{\tau}(n) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle, \tag{15}$$

where $\varphi(\mathbf{x}) = [\sqrt{|\mathbb{S}^{d-2}| \hat{\tau}(n)} Y_{n,j}(\mathbf{x})]_{n,j}$. Furthermore, we define $\tilde{\mathbf{w}} = \frac{\mathbf{w}}{\|\mathbf{w}\|_2}$ so that $\tilde{\mathbf{w}} \in \mathbb{S}^{d-1}$, and the function $\tilde{\sigma}_{\|\mathbf{w}\|_2}(\mathbf{x}^\top \tilde{\mathbf{w}}) := \sigma(\mathbf{x}^\top \mathbf{w})$. Then we can expand the function $\tilde{\sigma}_{\|\mathbf{w}\|_2}$ using the Funk-Hecke formula,

$$\sigma(\mathbf{x}^\top \mathbf{w}) = \sum_{n=0}^{\infty} |\mathbb{S}^{d-2}| \sum_{j=1}^{N(d,n)} Y_{n,j}(\mathbf{x}) Y_{n,j}(\tilde{\mathbf{w}}) \hat{\tilde{\sigma}}_{\|\mathbf{w}\|_2}(n) = \langle \varphi(\mathbf{x}), \zeta(\mathbf{w}) \rangle, \tag{16}$$

where $\zeta(\mathbf{w}) = [\sqrt{|S_{d-2}| / \hat{\tau}(n)} \hat{\tilde{\sigma}}_{\|\mathbf{w}\|_2}(n) Y_{n,j}(\tilde{\mathbf{w}})]_{n,j}$, and $\hat{\tilde{\sigma}}_{\|\mathbf{w}\|_2}(n)$ is the integral in Funk-Hecke formula for $\tilde{\sigma}_{\|\mathbf{w}\|_2}$. Because the spherical harmonics $\{Y_{n,j}\}_{n,j}$ are complete orthonormal basis

[1] We use $\hat{}$ to represent this integral operation hereafter.

of the function space on $\mathbb{S}^{d-1}$, the RKHS inner product can be computed as,

$$\begin{aligned}\langle \sigma(\mathbf{x}^\top \mathbf{w}), \sigma(\mathbf{x}^\top \mathbf{w}'))\rangle_{\mathcal{H}} = \langle \zeta(\mathbf{w}), \zeta(\mathbf{w}')\rangle &= |\mathbb{S}^{d-2}| \sum_{n=0}^{\infty} \frac{\hat{\tilde{\sigma}}_{\|\mathbf{w}\|_2}(n) \hat{\tilde{\sigma}}_{\|\mathbf{w}'\|_2}(n)}{\hat{\tau}(n)} \sum_{j=1}^{N(d,n)} Y_{n,j}(\tilde{\mathbf{w}}) Y_{n,j}(\tilde{\mathbf{w}'}) \\ &= \frac{|\mathbb{S}^{d-2}|}{|\mathbb{S}^{d-1}|} \sum_{n=0}^{\infty} \frac{\hat{\tilde{\sigma}}_{\|\mathbf{w}\|_2}(n) \hat{\tilde{\sigma}}_{\|\mathbf{w}'\|_2}(n)}{\hat{\tau}(n)} N(d,n) P_n^d(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{w}'}). \end{aligned} \tag{17}$$

where we used the Addition formula again in the last equality. Therefore, we have derived the RKHS inner products for the proposed variational features. Because the associated Legendre polynomials can be obtained using standard libraries, such as Scipy (Virtanen et al., 2020), what we need are only one-dimensional integrals for $\kappa, \tilde{\sigma}_{\|\mathbf{w}\|_2}, \tilde{\sigma}_{\|\mathbf{w}'\|_2}$. Furthermore, we can use a truncated summation $n = 0, ..., n_{max}$ to approximate the infinite summation.

3.3. Unifying Radial Basis Function and Feed Forward Neural Networks

A radial basis function (RBF) network (Chen et al., 1991) implements a mapping $f : \mathbb{R}^d \to \mathbb{R}$,

$$f_{rbf}(\mathbf{x}) = \sum_{m=1}^{M} \mathbf{a}_i \phi\left(\|\mathbf{x} - \mathbf{w}_i\|\right), \tag{18}$$

where $\{\mathbf{w}_i\}_{m=1}^M$ are RBF centers and ϕ is a given function. The hidden units depend on the radial distances, thus they are called radial basis function networks. Specifically, when $\phi\left(\|\mathbf{x} - \mathbf{w}_i\|\right) = k(\mathbf{x}, \mathbf{w}_i)$ is an instantiation from a stationary kernel k, the network is equivalently written as $f_{rbf}(\mathbf{x}) = \sum_{i=1}^m \mathbf{a}_i k(\mathbf{x}, \mathbf{w}_i)$. For example, $\phi(t) = e^{-t^2/2}$ is associated to the Squared Exponential kernel.

Feed forward neural networks (FFN) have been seen parallel with RBF networks. However, our variational features provide a unified perspective: both FFN and RBF networks are kernel regression models. Specifically, a FFN is represented as, $f_{ffn}(\mathbf{x}) = \sum_{i=1}^m \mathbf{a}_i \sigma\left(\mathbf{w}_i^\top \mathbf{x}\right)$. Treating each hidden unit $z_i := \sigma\left(\mathbf{w}_i^\top \cdot\right)$ as a variational feature for a dot-product kernel $k(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}^\top \mathbf{x}')$, then the network is equivalently written as $f_{ffn}(\mathbf{x}) = \sum_{i=1}^m \mathbf{a}_i k(\mathbf{x}, z_i)$ as well. Therefore, both the RBF networks and the feed-forward neural networks are instantiations of kernel regressions. If we regularize the data loss with the RKHS norm $\|f\|_{\mathcal{H}}^2 = \mathbf{a}^\top \mathbf{K}_{\mathbf{zz}} \mathbf{a}$, then we result in the kernel ridge regression, where $\mathbf{K}_{\mathbf{zz}}$ is the kernel matrix between inducing points,

$$[\mathbf{K}_{\mathbf{zz}}]_{ij} = k(z_i, z_j) = \langle \sigma(\mathbf{w}_i^\top \cdot), \sigma(\mathbf{w}_j^\top \cdot)\rangle_{\mathcal{H}}, \tag{19}$$

3.4. Nystrom Approximation Residuals as Predictive Variances

We can use the proposed variational features for stochastic variational Gaussian process (SVGP) (Titsias, 2009; Hensman et al., 2015) as well. Denote by $\mathcal{N}(\mathbf{K}_{\mathbf{zz}}\mathbf{a}, \mathbf{S})$ the variational posterior for inducing points $\mathbf{u}$, and $\mathbf{k}_{\mathbf{zx}} = [\sigma(\mathbf{w}_1^\top \mathbf{x}), ..., \sigma(\mathbf{w}_M^\top \mathbf{x})]^\top$, the posterior predictive mean and variance can be written as,

$$\mu(\mathbf{x}) = \mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{K}_{\mathbf{zz}} \mathbf{a} = \sum_{m=1}^{M} \mathbf{a}_m \sigma(\mathbf{w}_m^\top \mathbf{x}) = f(\mathbf{x}), \tag{20}$$

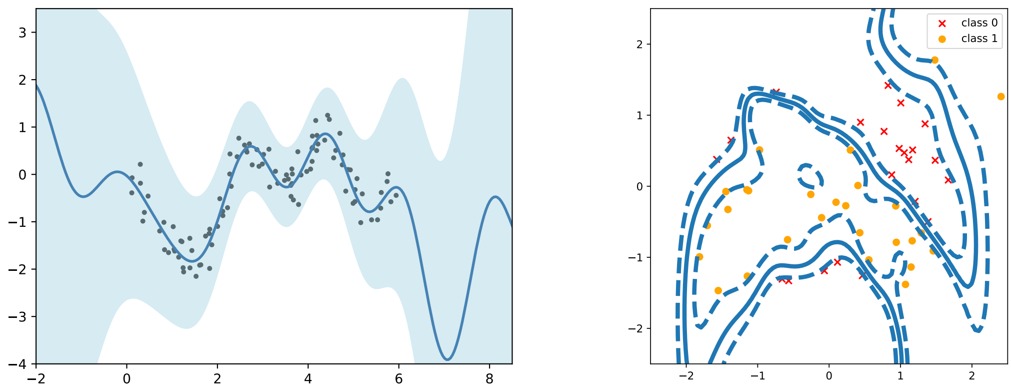
$$\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{zx}} + \mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{S} \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{zx}}, \tag{21}$$

We observe that the predictive mean function resembles the feed forward neural network f. In this way, the weights in the first layer parameterize inducing locations and the weights in the second layer parameterize the linear coefficients.

Instead of training $(\mathbf{a}, \mathbf{S})$ from scratch using SVGP objectives, we propose to directly generate predictive variances from a post-trained network f. Suppose we have obtained a M-hidden-unit neural network f trained using the standard losses, such as the squared loss or the cross entropy. The neurons $\sigma(\mathbf{w}_1^\top \mathbf{x}), ..., \sigma(\mathbf{w}_M^\top \mathbf{x})$ are inducing points for the kernel k, whose $\mathbf{k}_{\mathbf{zx}}$ and $\mathbf{K}_{\mathbf{zz}}$ can be obtained directly. We compute the Nystrom approximation residual of the inducing points, and use it as the predictive variance,

$$\hat{\sigma}^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{zx}}, \tag{22}$$

The Nystrom approximation residual is a lower bound of $\sigma^2(\mathbf{x})$. However, since $\mathbf{S}$ is usually small, $\hat{\sigma}^2(\mathbf{x})$ is an accurate estimate of $\sigma^2(\mathbf{x})$. Here we present experimental results for dot-product kernels on $\mathbb{R}$ and $\mathbb{S}^2$, where we firstly train the one-hidden-layer network, and then output the Nystrom approximation residuals as the predictive variances.



(a) $\tau = \cosh, \sigma = \cos, \mathcal{X} = \mathbb{R}$

(b) $\tau = ntk_{\text{erf}}, \sigma = \text{erf}, \mathcal{X} = \mathbb{S}^2$

Figure 1: Visualizing the network mean and the predictive variances (Nystrom approximation residuals). (a) We used the dot-product kernel $k(x, x') = \cosh(xx')$ and the activation $\sigma = \cos$, then the RKHS inner product function $\rho = \cosh$. The predictive mean is the prediction of a 20-hidden-unit network, and the predictive variance is the Nystrom residual $\hat{\sigma}^2(x)$ from the trained network. (b) We used the neural tangent kernel corresponding to one-hidden-layer networks with Erf activations. The RKHS inner products can be computed as we introduced in Subsection 3.2. For any input-target pair $((x_1, x_2), y)$ in the training set, $y \in \{0, 1\}$. We normalized $x_1 \in [0, \pi], x_2 \in [0, \frac{\pi}{2}]$, and embeded it on the sphere as $(\sin(x_1)\cos(x_2), \sin(x_1)\sin(x_2), \cos(x_1)) \in \mathbb{S}^2$. We trained a 100-hidden-unit neural network using the squared loss, which outputs the predictive mean $\mu(\mathbf{x})$. Given the trained network, we computed the Nystrom approximation residual for predictive variances $\hat{\sigma}^2(\mathbf{x})$. In the figure, the solid line represents $\mu(\mathbf{x}) = 0.5$ and the dashed lines represent $\mu(\mathbf{x}) \pm \hat{\sigma}(\mathbf{x}) = 0.5$.

Acknowledgments

We thank Yuhao Zhou for constructive discussions on the paper. SS was supported by the Connaught Fellowship.

References

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332, 2019.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

David R Burt, Carl Edward Rasmussen, and Mark van der Wilk. Variational orthogonal features. *arXiv preprint arXiv:2006.13170*, 2020.

Sheng Chen, Colin FN Cowan, and Peter M Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on neural networks*, 2(2): 302–309, 1991.

Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(Dec): 2153–2175, 2005.

Vincent Dutordoir, Nicolas Durrande, and James Hensman. Sparse gaussian processes with spherical harmonic features. *arXiv preprint arXiv:2006.16649*, 2020.

Jean Gallier. Notes on spherical harmonics and linear representations of lie groups. *preprint*, 2009.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. When do neural networks outperform kernel methods? *Advances in Neural Information Processing Systems*, 33, 2020.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360, 2015.

James Hensman, Nicolas Durrande, and Arno Solin. Variational fourier features for gaussian processes. *The Journal of Machine Learning Research*, 18(1):5537–5588, 2017.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

Mohammad Emtiyaz Khan, Alexander Immer, Ehsan Abedi, and Maciej Jan Korzepa. Approximate inference turns deep networks into gaussian processes. In *33rd Conference on Neural Information Processing Systems*, page 1751. Neural Information Processing Systems Foundation, 2019.

Miguel Lázaro-Gredilla and Anibal Figueiras-Vidal. Inter-domain gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, pages 1087–1095, 2009.

Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations*, 2018.

Mitsuo Morimoto. *Analytic functionals on the sphere.* American Mathematical Society, 1998.

Claus Müller. *Analysis of spherical symmetries in Euclidean spaces*, volume 129. Springer Science & Business Media, 2012.

Radford M Neal. *Bayesian Learning for Neural Networks.* PhD thesis, University of Toronto, 1995.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008.

Alex J Smola, Zoltan L Ovari, and Robert C Williamson. Regularization with dot-product kernels. In *Advances in neural information processing systems*, pages 308–314, 2001.

Sir William Thomson and Peter Guthrie Tait. *Treatise on natural philosophy.* 1888.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17 (3):261–272, 2020.