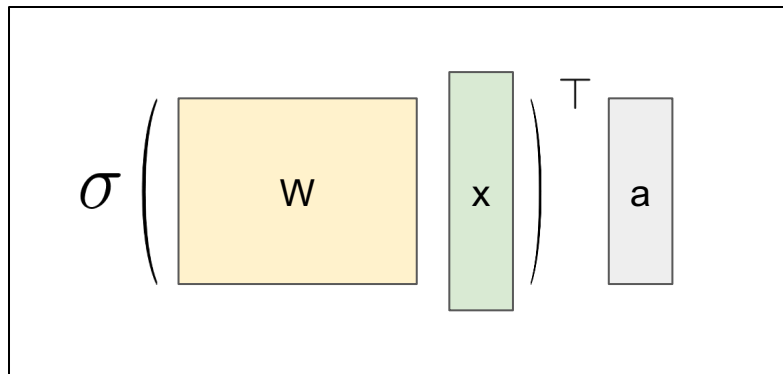


Neural Networks as Inter-Domain Inducing Points

Shengyang Sun*, Jiaxin Shi*, Roger Grosse



The diagram illustrates a neural network layer operation. It features a large yellow square labeled 'w' representing a weight matrix. To its right is a green vertical rectangle labeled 'x' representing an input vector. These two are enclosed within large parentheses, with a sigma symbol (σ) to the left, indicating an activation function. To the right of the parentheses is a grey vertical rectangle labeled 'a' representing the output vector. A superscript 'T' is positioned above the output vector 'a', indicating a transpose operation.

$$\sigma \left(w x \right)^T a$$



Existing Probabilistic Perspectives on Neural Networks

Infinite-width neural networks at initialization are Gaussian processes (Neal 92, Lee et al. 18)

$$f(\mathbf{x}) = \sum_{m=1}^M a_m \sigma(\mathbf{w}_m^\top \mathbf{x}) \quad w_{mj} \sim N(0, \sigma_w^2)$$

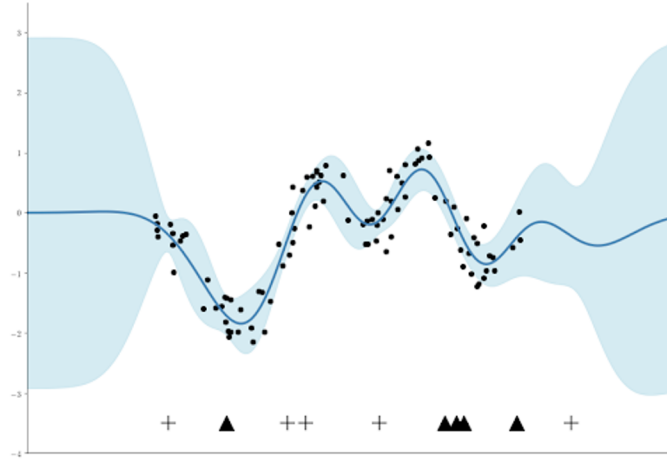
Infinite-width neural networks at training are Gaussian processes (NTK, Jacot et al. 18)

$$\Theta^{(L)}(x, y) := (\nabla f_\theta(x))^T \nabla f_\theta(y)$$

- Relies heavily on the infinite-width assumption.
- Ignores the importance of individual weights.
- Performance fails to match NNs with standard training.



Sparse Gaussian Processes



Sparse GP predictive distribution

Inducing Points (Z): A small number of inputs summarizing the training data



(Shi, Titsias, Mnih, 20)

Sparse GPs and Two-Layer NNs

Predictive mean of Sparse GP

$$\mu(\mathbf{x}) = k(\mathbf{Z}, \mathbf{x})^\top \mathbf{K}_{\mathbf{z}\mathbf{z}}^{-1} \mathbf{m}$$

Two-layer Neural Networks

$$f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x})^\top \mathbf{a}$$



Sparse GPs and Two-Layer NNs

Predictive mean of Sparse GP

$$\mu(\mathbf{x}) = k(\mathbf{Z}, \mathbf{x})^\top \mathbf{K}_{\mathbf{z}\mathbf{z}}^{-1} \mathbf{m}$$

Two-layer Neural Networks

$$f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x})^\top \mathbf{a}$$

Nonlinear

Linear



Sparse GPs and Two-Layer NNs

Predictive mean of Sparse GP $\mu(\mathbf{x}) = k(\mathbf{Z}, \mathbf{x})^\top \mathbf{K}_{\mathbf{z}\mathbf{z}}^{-1} \mathbf{m}$

Two-layer Neural Networks $f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x})^\top \mathbf{a}$

Problem: Activations are not necessarily positive-type functions



Sparse GPs and Two-Layer NNs

Predictive mean of Sparse GP $\mu(\mathbf{x}) = k(\mathbf{Z}, \mathbf{x})^\top \mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{m}$

Two-layer Neural Networks $f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x})^\top \mathbf{a}$

- Inter-domain inducing point $z : \mathcal{X} \rightarrow \mathbb{R}$
- Variational Fourier Features¹ (VFF) generalizes the kernel function

$$k(z, \mathbf{x}) = z(\mathbf{x}), k(z, z') = \langle z, z' \rangle_{\mathcal{H}}$$



¹ (Hensman et al, 2017)

Sparse GPs and Two-Layer NNs

Predictive mean of Sparse GP

$$\mu(\mathbf{x}) = k(\mathbf{Z}, \mathbf{x})^\top \mathbf{K}_{\mathbf{z}\mathbf{z}}^{-1} \mathbf{m}$$



Two-layer Neural Networks

$$z_i(\mathbf{x}) = \sigma(\mathbf{w}_i^\top \mathbf{x}) = k(\mathbf{z}_i, \mathbf{x})$$

$$f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x})^\top \mathbf{a}$$

- Inter-domain inducing point $z : \mathcal{X} \rightarrow \mathbb{R}$
- Variational Fourier Features¹ (VFF) generalizes the kernel function

$$k(z, \mathbf{x}) = z(\mathbf{x}), k(z, z') = \langle z, z' \rangle_{\mathcal{H}}$$



¹ (Hensman et al, 2017)

Numerical Experiments

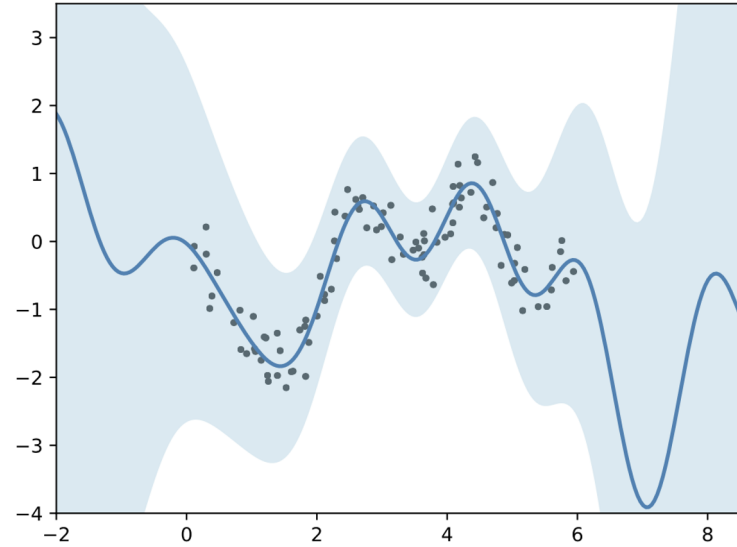
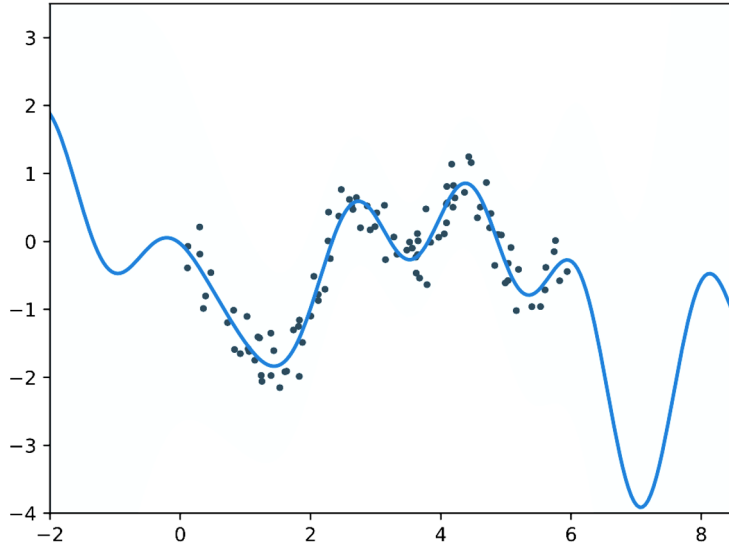
Uncertainty from post-trained NNs

1. Train a two-layer neural network by standard backprop.
2. After training, each hidden unit is an inter-domain inducing point.
3. Compute (approximate) predictive variance of the corresponding sparse GP:

$$\begin{aligned}\sigma^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{zx}} + \mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{S} \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{zx}} \\ &\approx k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathbf{zx}}^\top \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{zx}}\end{aligned}$$



Numerical Experiments



Uncertainty from post-trained NNs



Future Work

- Multi-layer neural networks
- Convolutional, Recurrent Structures
- How does this framework help us understand neural networks ?



Thanks

