Preliminary Research Idea (9): NNGP, Transformer and Dirichlet Process Combination

Professor Xu Yida
Machine Learning Knowledge Communicator

Prelude

First, a digression: Since 2019, I have been systematically reading NTK (Neural Tangent Kernel) related papers with a very capable Ph.D. student. At that time, we also organized a deep learning meetup in Sydney and were fortunate to invite Jacot+, one of the pioneers in the NTK field, to give a report. Although the popularity of NTK in the community seems not as high as when it was first proposed, in my opinion, it is still a very valuable theoretical research direction that deserves long-term exploration.

What is Neural Network Gaussian Process (NNGP)?

First, let's discuss what a Neural Network Gaussian Process+ (NNGP) is: When the width of each layer of a neural network tends to infinity, the function distribution induced by random weights converges to a Gaussian Process (GP). Its kernel function is jointly determined by the network structure, activation function+, and input distribution.

Basic Setup

Consider an L-layer fully connected network, with hidden layer widths $n_1, \ldots, n_L$, and weights and biases initialized independently and identically distributed:

$$f(x) = \frac{1}{n_L} W^{(L)} \phi\left(\cdots \phi\left(\frac{1}{n_1} W^{(1)} x + b^{(1)} \right) \cdots \right) + b^{(L)} \tag{1}$$

where ϕ is a nonlinearity (e.g., ReLU, erf, tanh), and weights $W_{ij}^{(\ell)}$ and biases $b_i^{(\ell)}$ follow a Gaussian distribution with appropriate variance. When all hidden layer widths $n_\ell \to \infty$ and the initialization distribution is fixed, the random function $f(\cdot)$ converges in distribution to a Gaussian process:

$$f(\cdot) \Rightarrow \mathcal{GP}(0, K_{\text{NNGP}}(\cdot,\cdot)) \tag{2}$$

What is the NNGP Kernel Function?

The NNGP kernel $K_{\text{NNGP}}(x, x')$ is defined recursively layer by layer. First, let $K^{(0)}(x, x') = \frac{1}{d}x^\top x'$ (or other given input covariance). For the ℓ-th layer, given $K^{(\ell-1)}$, it is defined as:

$$K^{(\ell)}(x, x') = \sigma_w^2 \mathbb{E}_{(u,v)\sim\mathcal{N}(0,\Sigma^{(\ell-1)})}[\phi(u)\phi(v)] + \sigma_b^2 \tag{3}$$

where

$$\Sigma^{(\ell-1)} = \begin{pmatrix} K^{(\ell-1)}(x,x) & K^{(\ell-1)}(x,x') \\ K^{(\ell-1)}(x',x) & K^{(\ell-1)}(x',x') \end{pmatrix} \tag{4}$$

For many common activation functions (such as ReLU, erf), the above expectation has an analytical form, thus yielding an explicit mapping from $K^{(\ell-1)}$ to $K^{(\ell)}$. After L layers, we get:

$$K_{\text{NNGP}}(x, x') = K^{(L)}(x, x') \tag{5}$$

Therefore, each combination of "structure + activation" corresponds to a specific GP kernel, similar to how an RBF kernel defines a classic GP.

Inference Based on NNGP

Given training data $\{(x_i, y_i)\}_{i=1}^N$ and additive Gaussian noise $y_i = f(x_i) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)$, inference under the NNGP prior is standard GP regression:

Where the prior is:

$$f(X) \sim \mathcal{N}(0, K_{\text{NNGP}}(X, X)) \tag{6}$$

Posterior (test point X_*):

$$f(X_*)|X, y \sim \mathcal{N}(\mu_*, \Sigma_*) \tag{7}$$

where

$$\mu_* = K_{\text{NNGP}}(X_*, X)(K_{\text{NNGP}}(X, X) + \sigma^2 I)^{-1} y \tag{8}$$

$$\Sigma_* = K_{\text{NNGP}}(X_*, X_*) - K_{\text{NNGP}}(X_*, X)(K_{\text{NNGP}}(X, X) + \sigma^2 I)^{-1} K_{\text{NNGP}}(X, X_*) \tag{9}$$

Therefore, NNGP provides a Bayesian non-parametric model corresponding to "infinite-width neural networks".

Extending NNGP to Transformers

One might guess that since NNGP became popular after 2018, there has been a series of works specifically deriving NNGP/NTK kernels for Transformers (self-attention architecture): The core idea is to treat the entire Transformer (including embedding layer, attention module, and MLP module) as a large random function at initialization. In the infinite width limit (e.g., attention head dimension, feed-forward layer dimension tending to infinity, with appropriate scaling), the model's outputs for different input sequences will jointly converge to a multivariate Gaussian distribution. For the outputs of two sequences X and X', their covariance defines a Transformer NNGP kernel $K_{\text{Trans}}(X, X')$.

Combining NNGP, Transformer and Dirichlet Process

The combination of NNGP, Transformer and Dirichlet Process+ opens up infinite possibilities for us. In fact, there are many ideas (today I'll just start with an opening, and I'll gradually complete them later): One research direction is DP over attention / heads / layers.

We view each attention head (or a complete attention block) as an infinitely wide neural network, corresponding to its own NNGP kernel, denoted as $K^{\text{Trans}}_{\text{head},k}$. We place a Dirichlet Process prior on the set of these "heads". Theoretically, this allows for countably infinite potential attention heads; through DP (e.g., stick-breaking construction+), we determine which heads and how many heads are actually used for specific tasks or datasets.

In function space, a typical form can be written as:

$$f(X) = \sum_{k=1}^{\infty} \pi_k f_k(X), \quad f_k \sim \mathcal{GP}(0, K_{\text{head},k}^{\text{Trans}}) \tag{10}$$

$$\{\pi_k\}_{k=1}^{\infty} \sim \text{stick-breaking DP} \tag{11}$$

Here π_k are the mixture weights obtained from the Dirichlet Process's stick-breaking construction. In this way, we obtain an infinite mixture model composed of Transformer-NNGP components, whose effective model complexity (e.g., "how many attention heads are actually active") is automatically adjusted by the Dirichlet Process.

This construction directly and naturally combines existing DP-GP mixture models with the NNGP/NTK derivations of Transformers, and may be an extensible direction.