Gaussian Process Bootstrapping Layer

Tetsuya Ishikawa

October 10, 2021

1 Theoretical details

The Gaussian process with random Fourier features can be regarded as a variant of a fully connected layer. We can design a new fully connected layer that can compute variance of it's intermediate features.

1.1 Gaussian process with random Fourier features

Let $\mathscr{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ be a training dataset where $\boldsymbol{x}_n$ is a n-th data and y_n is a label of n-th data. The Gaussian process with random Fourier features can be formulated as follows:

$$\boldsymbol{\phi}_{\boldsymbol{x}_n} = f(\boldsymbol{W}\boldsymbol{x}_n), \tag{1}$$

$$\boldsymbol{P} = \sum_{n=1}^{N} \boldsymbol{\phi}_{\boldsymbol{x}_n} \boldsymbol{\phi}_{\boldsymbol{x}_n}^\top, \tag{2}$$

$$m(\boldsymbol{x}_i) = \frac{1}{\sigma^2} \boldsymbol{y}^\top \boldsymbol{\Phi}^\top \left(\boldsymbol{I} - (\boldsymbol{P} + \sigma^2 \boldsymbol{I})^{-1} \right) \boldsymbol{\phi}_{\boldsymbol{x}_i}, \tag{3}$$

$$v(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}_{\boldsymbol{x}_i}^\top \left\{ \boldsymbol{I} - \frac{1}{\sigma^2} \boldsymbol{P} \left(\boldsymbol{I} - (\boldsymbol{P} + \sigma^2 \boldsymbol{I})^{-1} \right) \right\} \boldsymbol{\phi}_{\boldsymbol{x}_j}, \tag{4}$$

where f is a non-linear function and $\boldsymbol{W}$ is a random matrix derived from a kernel function of the Gaussian process.

1.2 Analogy to a fully connected layer

The formula $f(\boldsymbol{W}\boldsymbol{x})$ looks like a fully connected layer of neural network with activation function. For example, if the kernel function is RBF kernel $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2\right)$, then $\boldsymbol{\phi}_{\boldsymbol{x}_n}$ can be written as

$$\boldsymbol{\phi}_{\boldsymbol{x}_n} = \begin{pmatrix} \cos \boldsymbol{W}\boldsymbol{x}_n \\ \sin \boldsymbol{W}\boldsymbol{x}_n \end{pmatrix}, \quad \boldsymbol{W} \sim \mathscr{N}(0, \sigma^2). \tag{5}$$

In this case, $\boldsymbol{W}\boldsymbol{x}_n$ corresponds to a fully connected layer and the circuler functions correspond to an activation function.

On the other hand, $v(\boldsymbol{x}_n, \boldsymbol{x}_n)$ correspond to the variance of input data $\boldsymbol{x}_n$. Notable point is that the $v(\boldsymbol{x}_n, \boldsymbol{x}_n)$ is not depend on the label y_n, in other words, variance $v(\boldsymbol{x}_n, \boldsymbol{x}_n)$ is the same for any label. See the equation (4).

1.3 Gaussian process bootstrapping layer

Bacause of the independence of the variance and $v(\boldsymbol{x}_n, \boldsymbol{x}_n)$ and the label y_n mentioned in the previous subsection, we can replace the expectation prediction part as a identity function (theoretically, this situation correspond that $y_n = \boldsymbol{\phi}_{\boldsymbol{x}_n}$). See the figure 1. Therefore, we've got a new layer that can predict variance of intermediate features by replacing the random Fourier features as a fully connected layer, and expectation prediction as a identity function.

Gaussian process bootstrapping layer is a layer to add noises to the intermediate features where the variance of the noises is the variance of the intermediate features.

1.4 Psuedo code of GPB layer

The algorithm 1 is the pseudo code of the GPB layer.

References

[1] C. Rasmussen and C. Williams, “Gaussian Processes for Machine Learning”, MIT Press, 2006.

[2] A. Rahimi and B. Recht, "Random Features for Large-Scale Kernel Machines", NIPS, 2007.

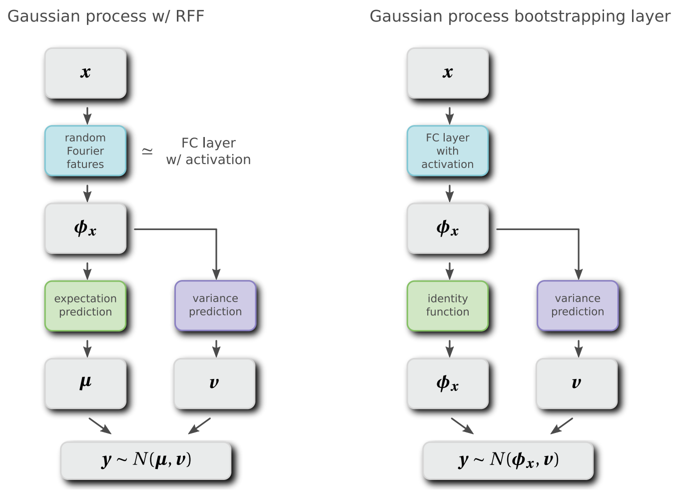

Figure 1: Illustration of the analogy between GP w/ RFF and GPB layer

Algorithm 1 Gaussian process bootstrapping layer

Input

$\boldsymbol{X}$: tensor with shape (N, C)

Output

$\boldsymbol{Y}$: tensor with shape (N, C)

Hyperpatameters

σ: standard deviation of measurement error

α: coefficient of exponential moving average

s: number of steps to skip bootstrapping

function GAUSSIAN PROCESS BOOTSTRAPPING LAYER(X, α, σ)

`# Update matrix P with exponential moving average.`

$\boldsymbol{P} = \alpha \boldsymbol{X}^\top \boldsymbol{X} + (1-\alpha)\boldsymbol{P}$

`# Compute matrix M.`

$\boldsymbol{M} = \boldsymbol{I} - \frac{1}{\sigma^2}\boldsymbol{P}\left(\boldsymbol{I} - (\boldsymbol{P} + \sigma^2 * \boldsymbol{I})^{-1}\boldsymbol{P}\right)$

`# Compute variance v[n].`

for n **in** $[0, N)$:

$v[n] = X[n,:]^\top M X[n,:]$

`# Add perturbation to the input tensor X.`

for n **in** $[0, N)$:

$Y[n,:] = X[n,:] + \sqrt{v[n]}\,\big(\text{sampling from normal distribution with shape } (1, C)\big)$

end function