

# 1. Deep Multi-Fidelity Gaussian Process Prediction

## 1.1 Brief Background

In daily life, we often encounter situations where it is difficult to obtain high-fidelity data during the training phase of some data predictions, such as changes in the earth's atmosphere and heat conduction data. But for these experiments, we can use artificially simulated physical models to obtain a large amount of data. But these artificially simulated data often have a gap with the real situation, that is, low-fidelity data. How to train a model with a large number of low-fidelity data obtained by simulation combined with very few high-precision high-fidelity data so that the model has a high correct rate of prediction for high-fidelity data has become a problem worthy of research.

## 1.2 Method of Experiment

### 1.2.1 Deep Multi-fidelity Gaussian Processes

We mainly use this structure to conduct joint training on high and low fidelity data in order to obtain information between high and low fidelity information.

$$x \begin{bmatrix} f_1(h) \\ f_2(h) \end{bmatrix} \sim \mathcal{GP} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_1(h, h') & \rho k_1(h, h') \\ \rho k_1(h, h') & \rho^2 k_1(h, h') + k_2(h, h') \end{bmatrix} \right)$$

Where

$$x \mapsto h := h(x) \mapsto \begin{bmatrix} f_1(h(x)) \\ f_2(h(x)) \end{bmatrix}$$

$f_1(h(x))$  : establish a low fidelity system

$f_2(h(x))$  : establish a high fidelity system

$\mathcal{GP}$ : Gaussian Processes

$h(x)$ : Arbitrarily determined parameter data conversion

But in the follow-up research, I kept trying and making mistakes, transforming  $h(x)$  into a multilayer neural network.

$$h(x) := (h^L \circ \dots \circ h^1)(x)$$

where

$$h^\ell(z) = \sigma^\ell(w^\ell z + b^\ell)$$

### 1.2.2 Forecasting Process

$$\begin{bmatrix} f_1(h) \\ f_2(h) \end{bmatrix} \sim \mathcal{GP} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_{11}(h, h') & k_{12}(h, h') \\ k_{21}(h, h') & k_{22}(h, h') \end{bmatrix} \right)$$

where:  $k_{11} \equiv k_1$

$k_{12} \equiv k_{21} \equiv \rho k_1$

$k_{22} \equiv \rho^2 k_1 + k_2$

This can be used to obtain the prediction distribution of the replacement model of the high-fidelity system at the new test point  $x_*$ :

$$p(f_2(h(x_*)) \mid x_*, \mathbf{x}_1, \mathbf{f}_1, \mathbf{x}_2, \mathbf{f}_2)$$

Introduce a new test point  $x_*$  and find the joint density:

$$\begin{bmatrix} f_2(h(x_*)) \\ \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} 0 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} k_{22}(h_*, h_*) & k_{21}(h_*, \mathbf{h}_1) & k_{22}(h_*, \mathbf{h}_2) \\ k_{12}(\mathbf{h}_1, h_*) & k_{11}(\mathbf{h}_1, \mathbf{h}_1) & k_{12}(\mathbf{h}_1, \mathbf{h}_2) \\ k_{22}(\mathbf{h}_2, h_*) & k_{21}(\mathbf{h}_2, \mathbf{h}_1) & k_{22}(\mathbf{h}_2, \mathbf{h}_2) \end{bmatrix} \right)$$

where  $h_* = h(x_*)$ ,  $\mathbf{h}_1 = h(\mathbf{x}_1)$ ,  $\mathbf{h}_2 = h(\mathbf{x}_2)$

Introduce a new test point  $x_*$  and find the joint density:

$$p(f_2(h(x_*)) \mid x_*, \mathbf{x}_1, \mathbf{f}_1, \mathbf{x}_2, \mathbf{f}_2) = \mathcal{N}(K_* K^{-1} \mathbf{f}, k_{22}(h_*, h_*) - K_* K^{-1} K_*^T)$$

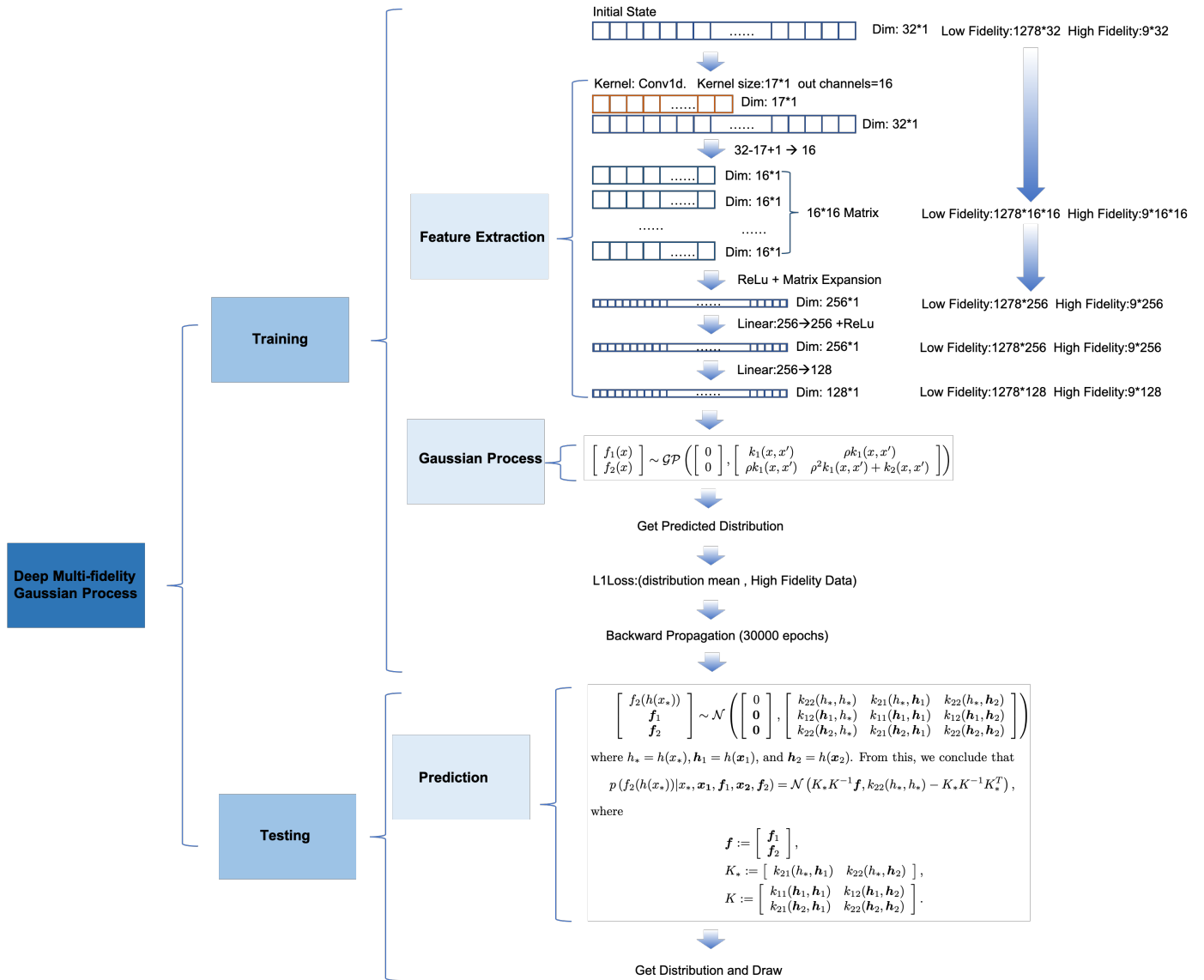
where:

$$\mathbf{f} := \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}$$

$$K_* := \begin{bmatrix} k_{21}(h_*, \mathbf{h}_1) & k_{22}(h_*, \mathbf{h}_2) \end{bmatrix}$$

$$K := \begin{bmatrix} k_{11}(\mathbf{h}_1, \mathbf{h}_1) & k_{12}(\mathbf{h}_1, \mathbf{h}_2) \\ k_{21}(\mathbf{h}_2, \mathbf{h}_1) & k_{22}(\mathbf{h}_2, \mathbf{h}_2) \end{bmatrix}$$

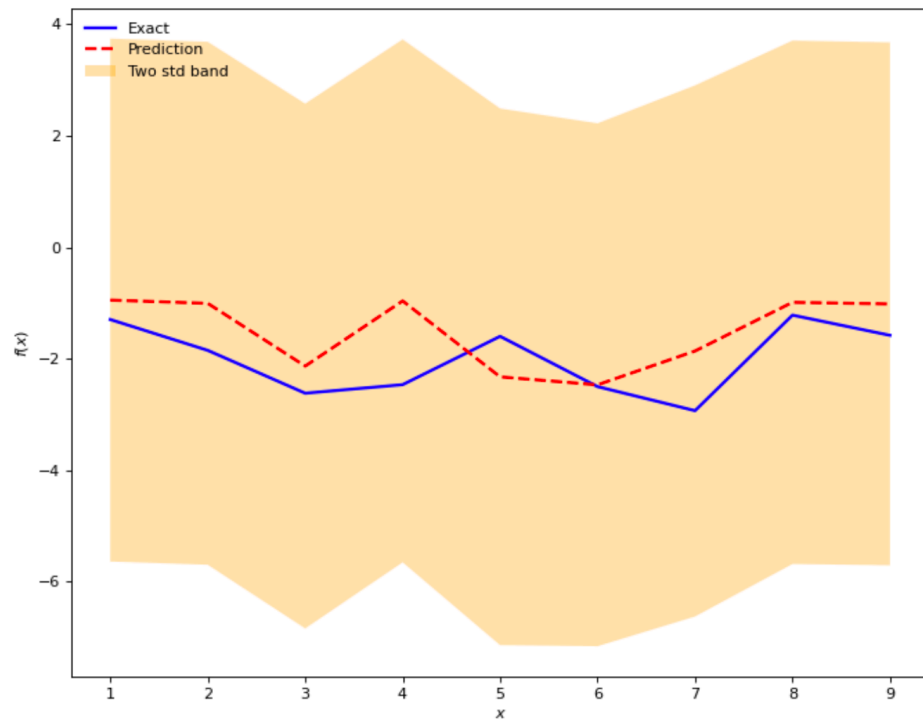
### 1.2.3 Project Framework



### 1.3 Training Effect Display

In my four versions of the improved code, evaluated using the R2 score, the accuracy of the predictions was eventually improved from a negative number to 0.62. All predictions fell within acceptable confidence intervals.

version1



version4

