

Kernel VAE: Gaussian Process Latent Variable Model on Deep Neural Network

Yuma Uchiumi^{*1}

¹Department of Information and Computer Science, Faculty of Science and Technology, Keio University, Japan

2019/10/11

Abstract

...

Contents

1	Introduction	2
2	ガウス過程と確率モデル	2
2.1	ガウス分布	2
2.2	ガウス過程 (GP)	2
2.3	ガウス過程回帰 (GPR)	3
2.4	ガウス過程潜在変数モデル (GP-LVM)	5
3	Deep Neural Network のカーネル法による近似	7
3.1	線形写像に対応するガウス過程	7
3.1.1	1 層の Feed Forward Neural Network	7
3.1.2	多層の Feed Forward Neural Network	8
3.2	カーネル法	11
3.2.1	正定値カーネルによる近似	11
3.2.2	多層の Feed Forward Neural Network	13

^{*}uchiumi@ailab.ics.keio.ac.jp

4	Kernel Variational AutoEncoder	14
4.1	変分推論	14
5	Experiments	14
6	Conclusion	14

1 Introduction

...

2 ガウス過程と確率モデル

2.1 ガウス分布

分布の再生性 任意の $n \in \mathbb{N}$ と $\{a_i, b_i \in \mathbb{R}\}_{i=1}^n$ に対して, 次が成り立つ.

$$X_i \sim i.i.d. \mathcal{N}(\mu_i, \sigma_i^2), \quad (i = 1, \dots, n) \quad (1)$$

$$\Rightarrow \sum_{i=1}^n a_i X_i + b_i \sim \mathcal{N}\left(\sum_{i=1}^n a_i \mu_i + b_i, \sum_{i=1}^n a_i^2 \sigma_i^2\right) \quad (2)$$

条件つき分布の計算 任意の2つのベクトル $\mathbf{f}_n \in \mathbb{R}^n, \mathbf{f}_m \in \mathbb{R}^m$ に対して,

$$\begin{pmatrix} \mathbf{f}_n \\ \mathbf{f}_m \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_n \\ \boldsymbol{\mu}_m \end{pmatrix}, \begin{bmatrix} \Sigma_{nn} & \Sigma_{nm} \\ \Sigma_{nm}^T & \Sigma_{mm} \end{bmatrix}\right) \quad (3)$$

ならば, 次が成り立つ.

$$\mathbf{f}_m | \mathbf{f}_n \sim \mathcal{N}(\boldsymbol{\mu}_{m|n}, \Sigma_{m|n}) \quad (4)$$

$$\text{where } \begin{cases} \boldsymbol{\mu}_{m|n} = \boldsymbol{\mu}_m + \Sigma_{nm}^T \Sigma_{nn}^{-1} (\mathbf{f}_n - \boldsymbol{\mu}_n) \\ \Sigma_{m|n} = \Sigma_{mm} - \Sigma_{nm}^T \Sigma_{nn}^{-1} \Sigma_{nm} \end{cases}$$

2.2 ガウス過程 (GP)

入力空間 \mathcal{X} 上の関数 $f: \mathcal{X} \rightarrow \mathbb{R}$ がガウス過程 (Gaussian Process, GP) に従うとは, \mathcal{X} 上の任意の n 点 $\{x_i\}_{i=1}^n$ に対して, ベクトル $\mathbf{f} = (f(x_1), \dots, f(x_n))^T \in \mathbb{R}^n$ が n 次元ガウス分布に従うことをいう. ここで, 確率変数 $\mathbf{f} \in \mathbb{R}^n$ が n 次

元ガウス分布に従う時、その確率密度関数 $p(\mathbf{f})$ は、平均関数 $m(\cdot)$ と共分散関数 $v(\cdot, \cdot)$ を用いて

$$p(\mathbf{f}) = \frac{1}{(2\pi)^{n/2} |V(\mathbf{f})|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{f} - m(\mathbf{f}))^T V(\mathbf{f})^{-1} (\mathbf{f} - m(\mathbf{f})) \right) \quad (5)$$

と定められる。ただし、 $V(\mathbf{f})$ は共分散 $v(\mathbf{f}_i, \mathbf{f}_j)$ を ij 要素にもつ共分散行列である。ゆえに、関数 $f: \mathcal{X} \rightarrow \mathbb{R}$ がガウス過程 (Gaussian Process, GP) に従うとき、その挙動は平均関数 $m(\cdot)$ と共分散関数 $v(\cdot, \cdot)$ によって定められ、これを以下のように記述する。

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), v(\cdot, \cdot)) \quad (6)$$

2.3 ガウス過程回帰 (GPR)

確率変数 $X \in \mathbb{R}^d, Y \in \mathbb{R}$ の実現値からなる n 個のデータサンプル $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ を用いて、 X の値から Y の値を推定するモデル $f: X \rightarrow Y$ を特定することを回帰問題という。すべての (\mathbf{x}, y) に対して、モデルの出力値 $f(\mathbf{x})$ と y との誤差を ε とおき、これが正規分布 $\mathcal{N}(0, \sigma^2)$ に従うと仮定すると回帰モデルは、

$$y = f(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2) \quad (7)$$

あるいは、正規分布の再生性より、

$$y|\mathbf{x} \sim \mathcal{N}(f(\mathbf{x}), \sigma^2) \quad (8)$$

となる。また一般の回帰問題において、データサンプル $\mathcal{D} = \{X_i, Y_i\}_{i=1}^n$ とモデル $f: X \rightarrow Y$ に対して下式が成り立つことから、これはモデル f の分布に関するベイズ推論へ拡張できる。

$$p(f|\mathcal{D}) = \frac{p(\mathcal{D}|f)p(f)}{p(\mathcal{D})}, \quad i.e. \quad p(f|Y, X) = \frac{p(Y|X, f)p(f)}{p(Y|X)} \quad (9)$$

上のモデルにおいて、関数 f がガウス過程に従う場合、これをガウス過程回帰という。たとえば、関数 f に対して、

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)) \quad (10)$$

を仮定すると、 $\mathbf{f}_n = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))^T$ と $\mathbf{y}_n = (y_1, \dots, y_n)^T$ に対して、

$$\mathbf{f}_n \sim \mathcal{N}(m(X_n), K(X_n, X_n)) \quad (11)$$

$$\mathbf{y}_n \sim \mathcal{N}(m(X_n), K(X_n, X_n) + \sigma^2 I_n) \quad (12)$$

が成り立つ。ただし、 $m(X_n) = (m(\mathbf{x}_1), \dots, m(\mathbf{x}_n))^T$, $K(X_n, K(X_n))_{ij} = k(f(\mathbf{x}_i), f(\mathbf{x}_j))$, I_n は $n \times n$ の単位行列とする。式 (10) は、式 (9) でモデル f の事前分布 $p(f)$ を定めることに対応する。さらに、式 (8) と正規分布の共役性より、ガウス過程回帰では、事前分布 $p(f)$ と事後分布 $p(f|Y, X)$ が共に正規分布に従うため、データサンプル $\mathcal{D} = \{X_i, Y_i\}_{i=1}^n$ に基づく平均関数 $m(\cdot)$ と共分散関数 $k(\cdot, \cdot)$ の行列計算 ($O(n^2)$) のみで事後分布の形状が求められる。

さらに、未知の m 個のデータ $\{\mathbf{x}_i\}_{i=n+1}^{n+m}$ に対して、対応する $\{y_i\}_{i=n+1}^{n+m}$ の同時分布 (予測分布) を求めることもできる。式 (10) より、

$$\begin{pmatrix} \mathbf{f}_n \\ \mathbf{f}_m \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} m(X_n) \\ m(X_m) \end{pmatrix}, \begin{bmatrix} K(X_n, X_n) & K(X_n, X_m) \\ K(X_n, X_m)^T & K(X_m, X_m) \end{bmatrix} \right) \quad (13)$$

が成り立つから、式 (4) より、 \mathbf{f}_m の \mathbf{f}_n に対する予測分布は、

$$\mathbf{f}_m | \mathbf{f}_n \sim \mathcal{N}(E[\mathbf{f}_m | \mathbf{f}_n], V[\mathbf{f}_m | \mathbf{f}_n]) \quad (14)$$

$$\text{where } \begin{cases} E[\mathbf{f}_m | \mathbf{f}_n] = m(X_m) + K(X_n, X_m)^T K(X_n, X_n)^{-1} (\mathbf{f}_n - m(X_n)) \\ V[\mathbf{f}_m | \mathbf{f}_n] = K(X_m, X_m) - K(X_n, X_m)^T K(X_n, X_n)^{-1} K(X_n, X_m) \end{cases}$$

となり、 \mathbf{f}_m の \mathbf{y}_n に対する予測分布は、

$$\mathbf{f}_m | \mathbf{y}_n \sim \mathcal{N}(E[\mathbf{f}_m | \mathbf{y}_n], V[\mathbf{f}_m | \mathbf{y}_n]) \quad (15)$$

$$\text{where } \begin{cases} E[\mathbf{f}_m | \mathbf{y}_n] = m(X_m) + K(X_n, X_m)^T (K(X_n, X_n) + \sigma^2 I_n)^{-1} (\mathbf{y}_n - m(X_n)) \\ V[\mathbf{f}_m | \mathbf{y}_n] = K(X_m, X_m) - K(X_n, X_m)^T (K(X_n, X_n) + \sigma^2 I_n)^{-1} K(X_n, X_m) \end{cases}$$

となる。よって、 \mathbf{y}_m の \mathbf{y}_n に対する予測分布は、

$$\mathbf{y}_m | \mathbf{y}_n \sim \mathcal{N}(E[\mathbf{y}_m | \mathbf{y}_n], V[\mathbf{y}_m | \mathbf{y}_n]) \quad (16)$$

$$\text{where } \begin{cases} E[\mathbf{y}_m | \mathbf{y}_n] = m(X_m) + K(X_n, X_m)^T (K(X_n, X_n) + \sigma^2 I_n)^{-1} (\mathbf{y}_n - m(X_n)) \\ V[\mathbf{y}_m | \mathbf{y}_n] = K(X_m, X_m) - K(X_n, X_m)^T (K(X_n, X_n) + \sigma^2 I_n)^{-1} K(X_n, X_m) + \sigma^2 I_m \end{cases}$$

となる。

2.4 ガウス過程潜在変数モデル (GP-LVM)

確率変数 $X \in \mathbb{R}^Q, Y \in \mathbb{R}^D$ に対して, Y の N 個のデータサンプル $\{\mathbf{y}_i\}_{i=1}^n$ から, 生成モデル $p(Y|X)$ を特定することを考える. このとき, Y を観測変数, X を潜在変数と呼ぶ. 観測変数の観測値 (計画行列) を $\mathbf{Y}^n = \mathbb{R}^{n \times D}$, それを表現する潜在変数の観測値 (計画行列) を $\mathbf{X}^n \in \mathbb{R}^{n \times Q}$ とおく. すべての $i \in \{1, \dots, N\}$ と $d \in \{1, \dots, D\}$ に対して, 関数 $f: \mathbb{R}^Q \rightarrow \mathbb{R}$ を用いて, 生成過程:

$$\mathbf{y}_{id} = f(\mathbf{x}_i) + \varepsilon_i, \quad \varepsilon_i \sim i.i.d. \mathcal{N}(0, \sigma_\varepsilon^2) \quad (17)$$

を仮定し, さらに関数 f に対してガウス過程

$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)) \quad (18)$$

を仮定すると,

$$p(\mathbf{y}_i | \mathbf{X}^n) = \prod_{d=1}^D p(\mathbf{y}_{id}) = \prod_{d=1}^D \mathcal{N}(\mathbf{y}_{id} | f(\mathbf{x}_i), \sigma_\varepsilon^2) \quad (19)$$

$$p(\mathbf{y}_d | \mathbf{X}^n) = \prod_{i=1}^n p(\mathbf{y}_{id}) = \mathcal{N}(\mathbf{y}_d | \mathbf{0}, \Sigma_{nn}) \quad (20)$$

となるから, 生成モデルの尤度は, 次式で与えられる.

$$p(\mathbf{Y}^n | \mathbf{X}^n) = \prod_{d=1}^D p(\mathbf{y}_d | \mathbf{X}^n) = \prod_{d=1}^D \mathcal{N}(\mathbf{y}_d | \mathbf{0}, \Sigma_{nn}) \quad (21)$$

$$= \prod_{d=1}^D \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{nn}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \mathbf{y}_d^T \Sigma_{nn}^{-1} \mathbf{y}_d\right) \quad (22)$$

$$= \frac{1}{(2\pi)^{\frac{Dn}{2}} |\Sigma_{nn}|^{\frac{D}{2}}} \exp\left(-\frac{1}{2} \text{tr}(\Sigma_{nn}^{-1} \mathbf{Y}^n \mathbf{Y}^{nT})\right) \quad (23)$$

ただし,

$$\Sigma_{nn} = \mathbf{K}_{nn} + \sigma_\varepsilon^2 \mathbf{I}_n \quad (24)$$

$$\mathbf{K}_{nn}(i, j) = k(\mathbf{x}_i, \mathbf{x}_j), \quad \forall (i, j) \quad (25)$$

とする.

最尤推定 \mathbf{Y}^n が与えられたときの \mathbf{X}^n の対数尤度は,

$$\log p(\mathbf{Y}^n|\mathbf{X}^n) = -\frac{Dn}{2} \log(2\pi) - \frac{D}{2} \log |\boldsymbol{\Sigma}_{nn}| - \frac{1}{2} \text{tr}(\boldsymbol{\Sigma}_{nn}^{-1} \mathbf{Y}^n \mathbf{Y}^{nT}) \quad (26)$$

となるから, 観測値によって固定された \mathbf{Y}^n の下で, これを目的関数として最大化すれば, \mathbf{X}^n の最尤推定量 $\hat{\mathbf{X}}_{ML}^n$ が求められる.

$$\hat{\mathbf{X}}_{ML}^n = \underset{\mathbf{X}^n}{\operatorname{argmax}} \log p(\mathbf{Y}^n|\mathbf{X}^n) \quad (27)$$

ここで, 対数尤度において \mathbf{X}^n に依存する変数は $\boldsymbol{\Sigma}_{nn}$ のみであることに注意する. 対数尤度の $\boldsymbol{\Sigma}_{nn}$ に対する勾配は, 次のように計算される.

$$\frac{\partial \log p(\mathbf{Y}^n|\mathbf{X}^n)}{\partial \boldsymbol{\Sigma}_{nn}} = \boldsymbol{\Sigma}_{nn}^{-1} \mathbf{Y} \mathbf{Y}^T \boldsymbol{\Sigma}_{nn}^{-1} - D \boldsymbol{\Sigma}_{nn}^{-1} \quad (28)$$

よって, 対数尤度の \mathbf{X}^n に対する勾配 $\partial \log p(\mathbf{Y}^n|\mathbf{X}^n)/\partial \mathbf{X}^n$ は, $\partial \boldsymbol{\Sigma}_{nn}/\partial \mathbf{X}^n = \partial \mathbf{K}_{nn}/\partial \mathbf{X}^n$ と連鎖律を用いて計算される.

$$\frac{\partial \log p(\mathbf{Y}^n|\mathbf{X}^n)}{\partial \mathbf{X}^n} = \frac{\partial \log p(\mathbf{Y}^n|\mathbf{X}^n)}{\partial \boldsymbol{\Sigma}_{nn}} \frac{\partial \boldsymbol{\Sigma}_{nn}}{\partial \mathbf{X}^n} \quad (29)$$

$$= (\boldsymbol{\Sigma}_{nn}^{-1} \mathbf{Y} \mathbf{Y}^T \boldsymbol{\Sigma}_{nn}^{-1} - D \boldsymbol{\Sigma}_{nn}^{-1}) \frac{\partial \mathbf{K}_{nn}}{\partial \mathbf{X}^n} \quad (30)$$

MAP 推定 \mathbf{X}^n の事前分布として,

$$p(\mathbf{X}^n) = \prod_{q=1}^Q p(\mathbf{x}_q) = \prod_{q=1}^Q \mathcal{N}(\mathbf{x}_q | \mathbf{0}, \sigma_x^2 I_N) \quad (31)$$

$$= \prod_{q=1}^Q \frac{1}{(2\pi\sigma_x^2)^{\frac{N}{2}}} \exp\left(-\frac{\mathbf{x}_q^T \mathbf{x}_q}{2\sigma_x^2}\right) \quad (32)$$

$$= \frac{1}{(2\pi\sigma_x^2)^{\frac{QN}{2}}} \exp\left(-\frac{1}{2\sigma_x^2} \text{tr}(\mathbf{X}^n \mathbf{X}^{nT})\right) \quad (33)$$

$$(34)$$

を仮定すると, ベイズの定理より, X の事後分布は,

$$p(X|Y) = \frac{1}{Z} p(Y|X) p(X) \quad (35)$$

となる．ただし， Z は規格化定数とする，よって， \mathbf{X}^n の事後確率は，

$$p(\mathbf{X}^n|\mathbf{Y}^n) \propto \log p(\mathbf{Y}^n|\mathbf{X}^n)p(\mathbf{X}^n) \quad (36)$$

$$= -\frac{Dn}{2} \log(2\pi) - \frac{D}{2} \log |\boldsymbol{\Sigma}_{nn}| - \frac{1}{2} \text{tr}(\boldsymbol{\Sigma}_{nn}^{-1} \mathbf{Y}^n \mathbf{Y}^{nT}) \quad (37)$$

$$- \frac{Qn}{2} \log(2\pi\sigma_x^2) - \frac{1}{2} \text{tr}(\mathbf{X}^n \mathbf{X}^{nT}) \quad (38)$$

となるから，これを目的関数として最大化すれば， \mathbf{X}^n の MAP 推定量は，次のように求められる．

$$\hat{\mathbf{X}}_{MAP}^n = \underset{\mathbf{X}^n}{\operatorname{argmax}} \log p(\mathbf{Y}^n|\mathbf{X}^n)p(\mathbf{X}^n) \quad (39)$$

3 Deep Neural Network のカーネル法による近似

3.1 線形写像に対応するガウス過程

3.1.1 1 層の Feed Forward Neural Network

ニューラルネットワークの全結合層や畳み込み層における推論計算は，線形写像と非線形の活性化関数の組み合わせによって構成される．ここで，ある層の入力ベクトルを $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^N$ ，出力ベクトルを $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^M$ ，線形写像に対応する変換行列を $\mathbf{W} \in \mathbb{R}^{M \times N}$ ，活性化関数を $\phi: \mathbb{R} \rightarrow \mathbb{R}$ とすると，各ノード y_i への推論処理は以下のように構成される．

$$y_i = \phi(z_i(\mathbf{x})), \quad z_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} = \sum_{j=1}^N w_{ij} x_j, \quad (i = 1, \dots, M) \quad (40)$$

ただし， w_{ij} は行列 \mathbf{W} の ij 要素とし， $\mathbf{w}_i = (w_{i1}, \dots, w_{iN})^T \in \mathbb{R}^N$ とする．なお，全結合層に置けるバイアスベクトルの追加に関しては， \mathbf{x} と \mathbf{W} に次元を 1 つ追加することで上式と等価となり，畳み込み層に置けるフィルタ演算も `im2col` によって上式と等価となることに注意する．ここで，重み行列 \mathbf{W} の各要素 w_{ij} に対して，

$$w_{ij} \text{ i.i.d. } \sim p_w(\cdot), \quad E[w_{ij}] = 0, \quad V[w_{ij}] = \sigma_w^2 \quad (41)$$

を仮定すると，任意の入力 \mathbf{x} に対して， $x_j \perp x_{j'} \ (j \neq j')$ だから，中心極限定理 (Central Limit Theorem) より，

$$z_i(\mathbf{x}) \sim \mathcal{N}(0, \sigma_w^2) \quad (N \rightarrow \infty) \quad (42)$$

が成り立つ．すなわち n 個の入力 $\{\mathbf{x}^{(i)}\}_{i=1}^n$ が与えられたとき， $\mathbf{z}_i = (z_i(\mathbf{x}^{(1)}), \dots, z_i(\mathbf{x}^{(n)}))^T \in \mathbb{R}^n$ は n 次元ガウス分布に従うから，

$$z_i(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)) \quad (43)$$

が与えられる．このガウス過程を用いて上述の推論処理を解く際には，共分散関数 (カーネル関数) $k(\cdot, \cdot) : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ を特定する必要がある．実際，任意の入力 $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ に対して，

$$k(\mathbf{x}, \mathbf{x}') = \text{Cov}[z_i(\mathbf{x}), z_i(\mathbf{x}')] = E_w[z_i(\mathbf{x})z_i(\mathbf{x}')] \quad (44)$$

$$= E_w \left[\left(\sum_{j=1}^N w_{ij} x_j \right) \left(\sum_{j=1}^N w_{ij} x'_j \right) \right] \quad (45)$$

$$= E_w \left[\sum_{j=1}^N \sum_{k=1}^N w_{ij} w_{ik} x_j x'_k \right] \quad (46)$$

$$= \sum_{j=1}^N \sum_{k=1}^N E_w[w^2] x_j x'_k \quad (47)$$

$$= \sigma_w^2 \sum_{j=1}^N \sum_{k=1}^N x_j x'_k \quad (48)$$

$$(49)$$

となり．さらに，任意の $\mathbf{x} = (x_1, \dots, x_N)^T \in \mathcal{X}$ に対して，

$$x_j \text{ i.i.d. } \sim p_x(\cdot), \quad E[x_j] = 0, \quad (i = 1, \dots, N) \quad (50)$$

を仮定すると，

$$E_{\mathbf{x}, \mathbf{x}'}[k(\mathbf{x}, \mathbf{x}')] = \sigma_w^2 \sum_{j=1}^N E[x_j x'_j] = \sigma_w^2 E[\mathbf{x}^T \mathbf{x}'] \quad (51)$$

となることに注意する．

3.1.2 多層の Feed Forward Neural Network

活性化関数 ϕ をもち L 層からなる Feed Forward Neural Network を考え， $l = 1, \dots, L$ に対して，第 l 層の活性化前の状態を $\mathbf{z}^{(l)}$ ，ノード数を $N^{(l)}$ とそ

れぞれおき，上の議論を多層に拡張する．すなわち，第 l 層の任意のノード値 $z_i^{(l)}(\cdot)$ に対して，ガウス過程

$$z_i^{(l)}(\cdot) \sim \mathcal{GP}(0, k^{(l)}(\cdot, \cdot)) \quad (52)$$

を仮定し，任意の入力 $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ に対して，この分散関数 $k^{(l)}$ を特定すればよい（図 1 参照）．上の議論から，任意の第 l 層に対して，カーネル関数 $k^{(l)}$ は

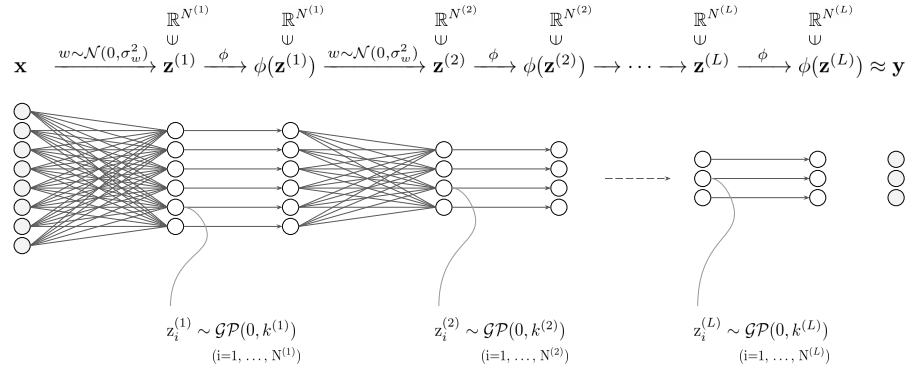


Figure 1: ガウス過程と DNN

次のように計算される.

$$k^{(l)}(\mathbf{x}, \mathbf{x}') = Cov[z_i^{(l)}(\mathbf{x}), z_i^{(l)}(\mathbf{x}')] \quad (53)$$

$$= E_w[z_i^{(l)}(\mathbf{x})z_i^{(l)}(\mathbf{x}')] \quad (54)$$

$$= E_w \left[\left(\sum_{j=1}^{N^{(l-1)}} w_{ij} \phi(z_j^{(l-1)}(\mathbf{x})) \right) \left(\sum_{j=1}^{N^{(l-1)}} w_{ij} \phi(z_j^{(l-1)}(\mathbf{x}')) \right) \right] \quad (55)$$

$$= E_w \left[\sum_{j=1}^{N^{(l-1)}} \sum_{k=1}^{N^{(l-1)}} w_{ij} w_{ik} \phi(z_j^{(l-1)}(\mathbf{x})) \phi(z_k^{(l-1)}(\mathbf{x}')) \right] \quad (56)$$

$$= \sum_{j=1}^{N^{(l-1)}} \sum_{k=1}^{N^{(l-1)}} E_w[w^2] \phi(z_j^{(l-1)}(\mathbf{x})) \phi(z_k^{(l-1)}(\mathbf{x}')) \quad (57)$$

$$= \sigma_w^2 \sum_{j=1}^{N^{(l-1)}} \sum_{k=1}^{N^{(l-1)}} \phi(z_j^{(l-1)}(\mathbf{x})) \phi(z_k^{(l-1)}(\mathbf{x}')) \quad (l \geq 2) \quad (58)$$

$$(59)$$

$$k^{(1)}(\mathbf{x}, \mathbf{x}') = Cov[z_i^{(1)}(\mathbf{x}), z_i^{(1)}(\mathbf{x}')] \quad (60)$$

$$= E_w[z_i^{(1)}(\mathbf{x})z_i^{(1)}(\mathbf{x}')] \quad (61)$$

$$= \sigma_w^2 \sum_{j=1}^{N^{(1)}} \sum_{k=1}^{N^{(1)}} x_j x'_k \quad (62)$$

さらに, 任意の $\mathbf{z}^{(l)} = (z^{(l)}(\mathbf{x}_1), \dots, z^{(l)}(\mathbf{x}_N))^T \in \mathbb{R}^N$ に対して,

$$z^{(l)}(\mathbf{x}_i) \text{ i.i.d. } \sim p_z(\cdot), \quad E[z^{(l)}(\mathbf{x}_i)] = 0, \quad (i = 1, \dots, N) \quad (63)$$

を仮定すると,

$$\phi(z^{(l)}(\mathbf{x}_i)) \text{ i.i.d. } \sim p_{\phi(z)}(\cdot), \quad E[\phi(z^{(l)}(\mathbf{x}_i))] = 0, \quad (i = 1, \dots, N) \quad (64)$$

だから,

$$E_{\mathbf{x}, \mathbf{x}'} \left[k^{(l)}(\mathbf{x}, \mathbf{x}') \right] \quad (65)$$

$$= E_{\mathbf{x}, \mathbf{x}'} \left[\sigma_w^2 \sum_{j=1}^{N^{(l-1)}} \sum_{k=1}^{N^{(l-1)}} \phi(z_j^{(l-1)}(\mathbf{x})) \phi(z_k^{(l-1)}(\mathbf{x}')) \right] \quad (66)$$

$$= \sigma_w^2 E_{\mathbf{z}^{(l-1)}, \mathbf{z}'^{(l-1)}} \left[\sum_{j=1}^{N^{(l-1)}} \phi(z_j^{(l-1)}) \phi(z_j'^{(l-1)}) \right] \quad (67)$$

$$= \sigma_w^2 E_{\mathbf{z}^{(l-1)}, \mathbf{z}'^{(l-1)}} \left[\boldsymbol{\phi}(\mathbf{z}^{(l-1)})^T \boldsymbol{\phi}(\mathbf{z}'^{(l-1)}) \right] \quad (68)$$

$$= \sigma_w^2 N^{(l-1)} E_{\mathbf{z}^{(l-1)}, \mathbf{z}'^{(l-1)} \sim \mathcal{GP}(0, k^{(l-1)})} \left[\phi(z^{(l-1)}) \phi(z'^{(l-1)}) \right] \quad (69)$$

となることに注意する.

3.2 カーネル法

3.2.1 正定値カーネルによる近似

式 (40) の推論処理をまとめて関数 $f_i: \mathbb{R}^N \rightarrow \mathbb{R}$ とおく.

$$y_i = f_i(\mathbf{x}) = \phi(\mathbf{w}_i^T \mathbf{x}) \quad (70)$$

ここで, 関数 f_i に対する内積 $f_i(\cdot)^T f_i(\cdot): \mathbf{x} \times \mathbf{x}' \mapsto \mathbb{R}$ を考える. 活性化関数 ϕ として ReLU を仮定して, 対応する指示関数

$$I(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases} \quad (71)$$

を定義すると,

$$f_i(\mathbf{x}) f_i(\mathbf{x}') = I(\mathbf{w}_i^T \mathbf{x}) I(\mathbf{w}_i^T \mathbf{x}') (\mathbf{w}_i^T \mathbf{x}) (\mathbf{w}_i^T \mathbf{x}') \quad (72)$$

$$= I\left(\sum_{j=1}^N w_{ij} x_j\right) I\left(\sum_{j=1}^N w_{ij} x'_j\right) \left(\sum_{j=1}^N w_{ij} x_j\right) \left(\sum_{j=1}^N w_{ij} x'_j\right) \quad (73)$$

となる. 中心極限定理 (Central Limit Theorem) より, 任意の $\mathbf{w} \in \mathbb{R}^N$ の関数 $C(\mathbf{w})$ に対して,

$$E \left[\frac{1}{M} \sum_{i=1}^M C(\mathbf{w}_i) \right] \rightarrow \int d\mathbf{w} \frac{\exp(-\frac{\|\mathbf{w}\|^2}{2})}{(2\pi\sigma_w^2)^{N/2}} C(\mathbf{w}) \quad (M \rightarrow \infty) \quad (74)$$

が満たされるから, \mathbf{x}, \mathbf{x}' を固定したとき,

$$\begin{aligned} E \left[\frac{1}{M} \sum_{i=1}^M f_i(\mathbf{x}) f_i(\mathbf{x}') \right] &= E \left[\frac{1}{M} \sum_{i=1}^M \mathbf{I}(\mathbf{w}_i^T \mathbf{x}) \mathbf{I}(\mathbf{w}_i^T \mathbf{x}') (\mathbf{w}_i^T \mathbf{x}) (\mathbf{w}_i^T \mathbf{x}') \right] \\ &\rightarrow \int d\mathbf{w} \frac{\exp(-\frac{\|\mathbf{w}\|^2}{2})}{(2\pi\sigma_w^2)^{N/2}} \mathbf{I}(\mathbf{w}^T \mathbf{x}) \mathbf{I}(\mathbf{w}^T \mathbf{x}') (\mathbf{w}^T \mathbf{x}) (\mathbf{w}^T \mathbf{x}') \quad (M \rightarrow \infty) \end{aligned} \quad (75)$$

が成り立つ. 式 (75) 右辺を整理すると, 以下の定理が導かれる.

定義 1 (*Arc-Cosine kernel*) 任意の 2 つのベクトル $\mathbf{x}, \mathbf{x}' \in \mathcal{X} \subset \mathbb{R}^N$ に対して,

$$\theta = \cos^{-1} \left(\frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\| \|\mathbf{x}'\|} \right) \quad (76)$$

$$J(\theta) = \sin \theta + (\pi - \theta) \cos \theta \quad (77)$$

とおき, *Arc-Cosine* カーネル

$$k^{(1)}(\mathbf{x}, \mathbf{x}') = \frac{1}{\pi} \|\mathbf{x}\| \|\mathbf{x}'\| J(\theta) \quad (78)$$

を定義する.

定理 1 各要素 w_{ij} が *i.i.d.* で, 平均が 0, 分散が σ_w^2 となるランダム行列 $\mathbf{W} \in \mathbb{R}^{M \times N}$ を考える. 活性化関数として *ReLU* 関数 ϕ を用いて, 1 層のニューラルネットワークに相当する関数 $f_i: \mathbb{R}^N \rightarrow \mathbb{R}$:

$$f_i(\mathbf{x}) = \phi(\mathbf{w}_i^T \mathbf{x}), \quad (i = 1, \dots, M) \quad (79)$$

を考える. このとき任意の $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^N$ に対して,

$$E \left[\frac{1}{M} \sum_{i=1}^M f_i(\mathbf{x}) f_i(\mathbf{x}') \right] \rightarrow \frac{1}{\sigma_w^N} k^{(1)}(\mathbf{x}, \mathbf{x}') \quad (M \rightarrow \infty) \quad (80)$$

が成り立つ.

定理 2 $\mathcal{X} \times \mathcal{X}$ 上の関数

$$k^{(1)}(\mathbf{x}, \mathbf{x}') = \frac{1}{\pi} \|\mathbf{x}\| \|\mathbf{x}'\| J(\theta) \quad (81)$$

は半正定値関数である.

定理 3 (カーネル関数存在定理) 任意の対称行列 $K \in \mathbb{R}^{n \times n}$ が半正定値ならば, データ空間 \mathcal{X} 上の n 点 $\{\mathbf{x}_i\}_{i=1}^n$ と, 特徴空間 $\mathcal{F} \subset \mathbb{R}^n$ 上の n 次元特徴ベクトル $\{f(\mathbf{x}_i)\}_{i=1}^n$ がそれぞれ存在して,

$$K_{ij} = \langle f(\mathbf{x}_i), f(\mathbf{x}_j) \rangle_{\mathcal{F}} = \sum_{k=1}^n f(\mathbf{x}_i)_k f(\mathbf{x}_j)_k \quad (82)$$

が成り立つ. さらに, $k(\mathbf{x}_i, \mathbf{x}_j) = K_{ij}$ とおけば, 半正定値関数 $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ が定義される.

定理 3 から, カーネル関数 $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ を定義せずとも, 与えられたデータサンプルから構成された半正定値対称行列を作ることによって, それが何らかのカーネル関数によるグラム行列に相当することが保証される. よって, 定理 1 と定理 2 より, 式 (81) を共分散関数にもつガウス過程が 1 層の Neural Network の近似となることがわかる.

3.2.2 多層の Feed Forward Neural Network

1 層の Feed Forward Neural Network に関する議論は, 正定値カーネルの線形性より, 容易に多層へと拡張することができる. 一般性を失うことなく, $L > 0$ 層からなる Feed Forward Neural Network を考え, 各層の線形写像に対応する重み行列 $\mathbf{W}^{(l)}$ に対して, 固定された σ_w^2 を与えて,

$$\forall l \in [1, L] \subset \mathbb{N}, \quad \mathbf{W}_{ij}^{(l)} \sim i.i.d. \mathcal{N}(0, \sigma_w^2) \quad (83)$$

を仮定する. 第 l 層のユニット数を $N^{(l)}$, 第 l 層の空間を $\mathcal{T}^{(l)}$, 第 l 層の出力を得る関数を $f^{(l)}: \mathcal{X} \rightarrow \mathcal{T}^{(l)}$ とおく. 任意の入力データ $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ に対して, Neural Network の第 l 層に対応する共分散関数 $k^{(l)}: \mathcal{T}^{(l)} \times \mathcal{T}^{(l)} \rightarrow \mathbb{R}$ は, 次の漸化式で求められる.

$$\begin{aligned} k^{(1)}(\mathbf{x}, \mathbf{x}') &= Cov \left[f^{(1)}(\mathbf{x}), f^{(1)}(\mathbf{x}') \right] \\ &= Cov \left[\phi(\mathbf{W}^{(1)}\mathbf{x}), \phi(\mathbf{W}^{(1)}\mathbf{x}') \right] \\ &= \frac{1}{\pi \sigma_w^{N^{(1)}}} \|\mathbf{x}\| \|\mathbf{x}'\| J(\theta^{(0)}) \end{aligned} \quad (84)$$

$$\begin{aligned} k^{(l+1)}(\mathbf{x}, \mathbf{x}') &= Cov \left[f^{(l+1)}(\mathbf{x}), f^{(l+1)}(\mathbf{x}') \right] \\ &= Cov \left[\phi(\mathbf{W}^{(l+1)} \dots \phi(\mathbf{W}^{(1)}\mathbf{x})), \phi(\mathbf{W}^{(l+1)} \dots \phi(\mathbf{W}^{(1)}\mathbf{x}')) \right] \\ &= \frac{1}{\pi \sigma_w^{N^{(l+1)}}} \sqrt{k^{(l)}(\mathbf{x}, \mathbf{x})} \sqrt{k^{(l)}(\mathbf{x}', \mathbf{x}')} J(\theta^{(l)}) \end{aligned} \quad (85)$$

ただし,

$$J(\theta^{(l)}) = \sin \theta^{(l)} + (\pi - \theta^{(l)}) \cos \theta^{(l)} \quad (86)$$

$$\theta^{(l)} = \begin{cases} \cos^{-1} \left(\frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\| \|\mathbf{x}'\|} \right) & (l = 0) \\ \cos^{-1} \left(\frac{k^{(l)}(\mathbf{x}, \mathbf{x}')}{\sqrt{k^{(l)}(\mathbf{x}, \mathbf{x})} \sqrt{k^{(l)}(\mathbf{x}', \mathbf{x}')}} \right) & (l \neq 0) \end{cases} \quad (87)$$

とする. 以上の結果から, 入力から第 l 層の各ユニットの出力値を得る関数 $f_i^{(l)} : \mathcal{X} \rightarrow \mathbb{R}$ は以下のガウス過程で近似できる.

$$f_i^{(l)}(\cdot) \sim \mathcal{GP}(0, k^{(l)}(\cdot, \cdot)), \quad (i = 1, \dots, N^{(l)}) \quad (88)$$

4 Kernel Variational AutoEncoder

前章での議論から, GP-LVM における潜在変数 X と観測関数 Y に対する確率モデル $p(Y|X)$ を, 多層の Feed Forward Neural Network に相当するモデル

$$y = f^{(l)}(\mathbf{x}) = \phi(\mathbf{w}^{(l)T} \phi(\mathbf{W}^{(l-1)} \dots \phi(\mathbf{W}^{(1)} \mathbf{x}))) \quad (89)$$

によって表現することを考える. すなわち,

$$k^{(l)}(\mathbf{x}, \mathbf{x}') = \langle f^{(l)}(\mathbf{x}), f^{(l)}(\mathbf{x}') \rangle_{\mathcal{F}^{(l)}} \quad (90)$$

に対応するガウス過程

$$f^{(l)}(\cdot) \sim \mathcal{GP}(0, k^{(l)}(\cdot, \cdot)) \quad (91)$$

を考える.

4.1 変分推論

5 Experiments

6 Conclusion

文献 [1] [2] [3] [4] [5] [6] [7] [8] [9]

References

- [1] Geoffrey E. Hinton and Radford M. Neal. Bayesian learning for neural networks. 1995.
- [2] Jaehoon Lee, Yasaman Bahri, Roman Novak, Sam Schoenholz, Jeffrey Pennington, and Jascha Sohl-dickstein. Deep neural networks as gaussian processes. 2018.
- [3] Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pp. 342–350. Curran Associates, Inc., 2009.
- [4] Neil D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS’03*, pp. 329–336, Cambridge, MA, USA, 2003. MIT Press.
- [5] Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *J. Mach. Learn. Res.*, Vol. 6, pp. 1783–1816, December 2005.
- [6] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pp. 1257–1264. MIT Press, 2006.
- [7] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, Vol. 5 of *Proceedings of Machine Learning Research*, pp. 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.
- [8] Michalis Titsias and Neil D. Lawrence. Bayesian gaussian process latent variable model. In Yee Whye Teh and Mike Titterton, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9 of *Proceedings of Machine Learning Research*, pp. 844–851, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

- [9] James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, pp. 282–290, Arlington, Virginia, United States, 2013. AUAI Press.