

Neural Process

- **1. Gaussian Process:** 使用“核函数”描述预测标签上的条件分布 $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X})$

↓
问题: 1. 核函数需要人为指定 2. 条件分布 $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X})$ 的计算复杂度高

- **2. Conditional Neural Process (CNP):** 直接使用神经网络 θ 描述上述条件分布 $p_\theta(\mathbf{Y}_T|\mathbf{X}_T, \mathbf{C})$

↓
问题: 未考虑和Target Set内样本点 $x^{(t)}$ 之间的“位置关系”，导致“预测结果单一”
(特别是Context Set内样本数量较少时)

- **3. Latent Neural Processes (LNP):** 引入隐变量 $\mathbf{z}$, 并使用“边缘化条件预测分布” $\int p_\theta(\mathbf{Y}_T|\mathbf{X}_T, \mathbf{C}, \mathbf{z})p_\theta(\mathbf{z}|\mathbf{C})d\mathbf{z}$

- 3.1 Maximum Likelihood-LNP (ML-LNP): 直接极大化“边缘似然” $\int p_\theta(\mathbf{Y}_T|\mathbf{X}_T, \mathbf{C}, \mathbf{z})p_\theta(\mathbf{z}|\mathbf{C})d\mathbf{z}$

↓
问题: 从隐变量 $\mathbf{z}$ 的先验分布 $p_\theta(\mathbf{z}|\mathbf{C})$ 中进行采样，往往采到很多对“边缘似然”没有贡献的样本

- 3.2 Variational Inference-LNP (VI-LNP): 从近似后验分布 $q_\theta(\mathbf{z}|\mathbf{C}, \mathbf{T})$ 中进行采样，极大化 $\text{ELBO}(\theta)$

↓
问题: 未考虑Context Set和Target Set内样本点的位置关系。

- **4. Attentive LNP (AttnLNP):** 利用 $\mathbf{X}_C$ 和 $\mathbf{X}_T$ 之间的“距离信息”对特征 $\mathbf{R}^{(c)}$ 进行“加权平均”

1. Gaussian Process (GP)

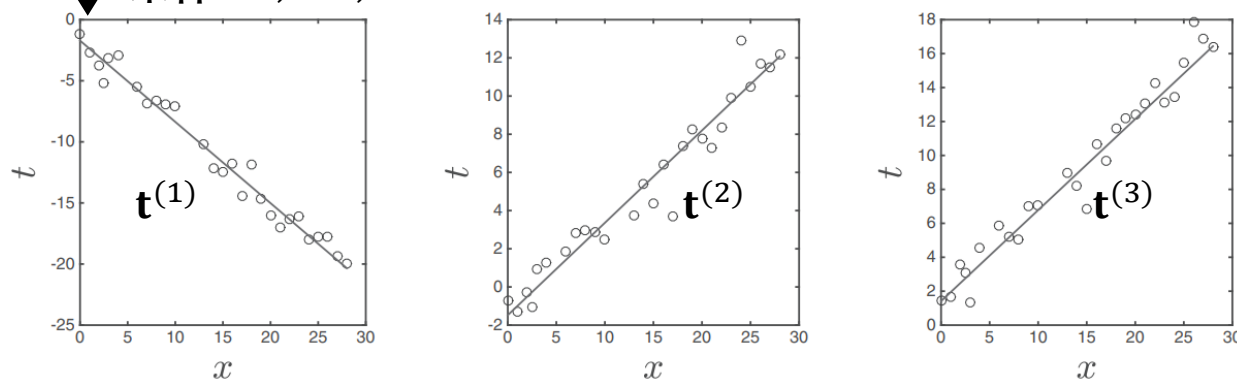
- 考虑一维贝叶斯线性回归例子: $t_i = \mathbf{w}^\top \mathbf{x}_i$

- 先验: $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; 0, \mathbf{I})$. 似然: $p(\mathbf{t}|\mathbf{w}, \mathbf{X}; \sigma^2) = \mathcal{N}(\mathbf{t}; \mathbf{X}\mathbf{w}, \sigma^2 \mathbf{I})$

↓ 采样 $\mathbf{w}_i$

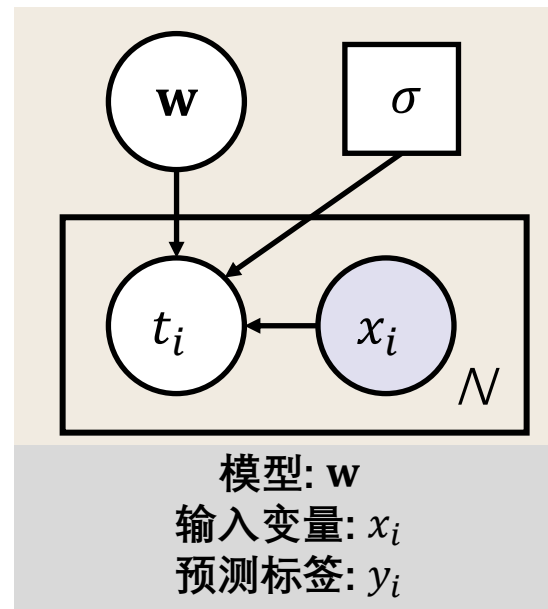
- 模型 $\mathbf{w}_i$ 的预测分布: $p(\mathbf{t}|\mathbf{w}_i, \mathbf{X}; \sigma^2) = \mathcal{N}(\mathbf{t}; \mathbf{X}\mathbf{w}_i, \sigma^2 \mathbf{I})$. →

↓ 采样 $\mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \mathbf{t}^{(3)}$



假定有预测N个样本,
则该分布为N维高斯分布

1. 均值向量为 $\mathbf{X}\mathbf{w}_i$
2. 协方差矩阵为“对角矩阵”
3. 预测分布与模型 $\mathbf{w}$ 有关



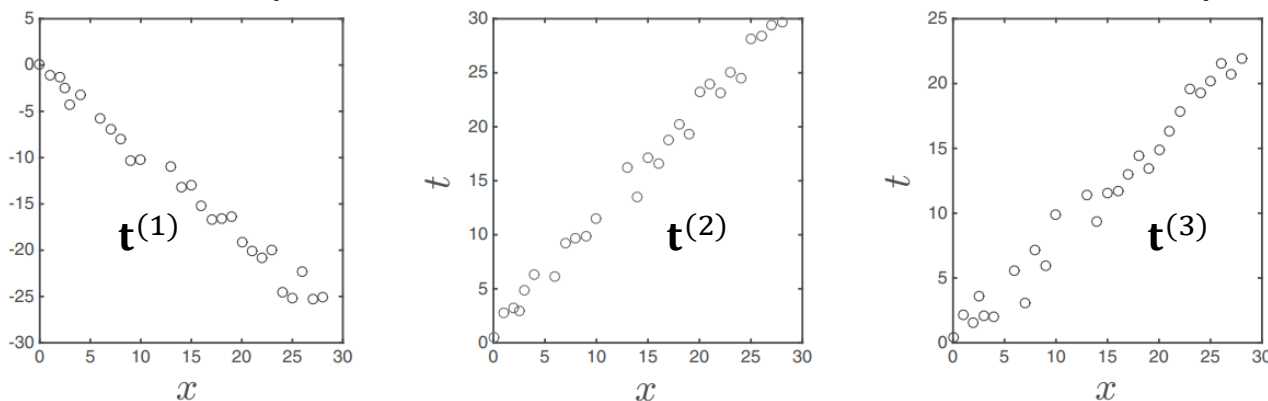
- 先验预测分布: $p(\mathbf{t}|\mathbf{X}; \sigma^2) = \int p(\mathbf{t}|\mathbf{w}, \mathbf{X}; \sigma^2) p(\mathbf{w}) d\mathbf{w} = \mathcal{N}(\mathbf{t}; \mathbf{0}, \sigma^2 \mathbf{I} + \mathbf{X}^\top \mathbf{X})$ →

直接
边缘化z

↓ 采样 $\mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \mathbf{t}^{(3)}$

虽然均值为0, 但是协方差矩阵不是对角矩阵, 因此, $t_i^{(1)}$ 采样的结果会影响 $t_{i+1}^{(1)}$ 的结果。

例如: $t_i^{(1)} = 5$, 则 $t_{i+1}^{(1)} \approx 5$



边缘似然

(因为“先验”和“似然”共轭, 因此该边缘似然可以直接得出)

1. 均值向量为0向量
2. 协方差矩阵为“非对角矩阵”
3. 预测分布与模型 $\mathbf{w}$ 无关

能否忽略模型参数 $\mathbf{w}$ 的表示, 直接描述“先验预测分布”?

Gaussian Process (GP)

1. Gaussian Process (GP)-参数化先验预测 $p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{C})$

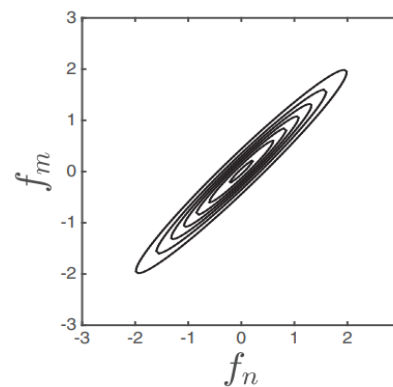
- **定义先验预测分布：** $p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{C})$ ，其中， $\mathbf{f}$ 相当于上述的预测标签 $\mathbf{t}$ ，预测分布的均值为0，协方差矩阵由核函数-“RBF (Radial Basis Function)”所确定。

- 核函数RBF: $c(x_n, x_m) = \alpha \exp(-\gamma(x_n - x_m)^2)$ ；衡量 x_n 和 x_m 的距离，距离越近，协方差越大，说明“关联性越强”；反之，“关联性越弱”。

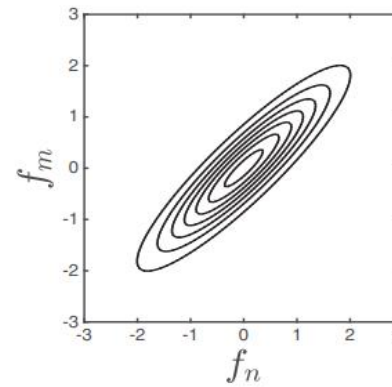
- 协方差矩阵: $\mathbf{C} = \begin{bmatrix} c(x_1, x_1) & \cdots & c(x_1, x_n) \\ \vdots & \ddots & \vdots \\ c(x_n, x_1) & \cdots & c(x_n, x_n) \end{bmatrix}$

采样 $\mathbf{f}^{(1)}, \mathbf{f}^{(2)}, \dots, \mathbf{f}^{(n)}$

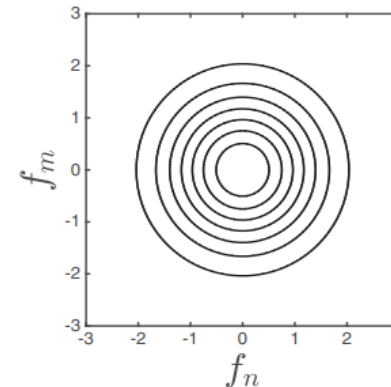
任意选出 f_n, f_m 组成
二维高斯分布



(a) $|x_n - x_m| = 0.5$

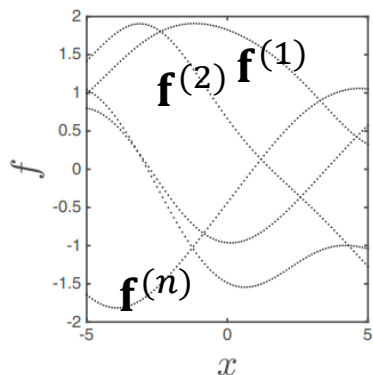


(b) $|x_n - x_m| = 1.0$

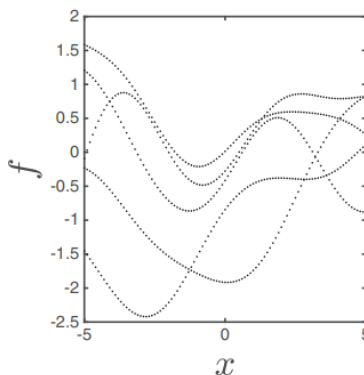


(d) $|x_n - x_m| = 10.0$

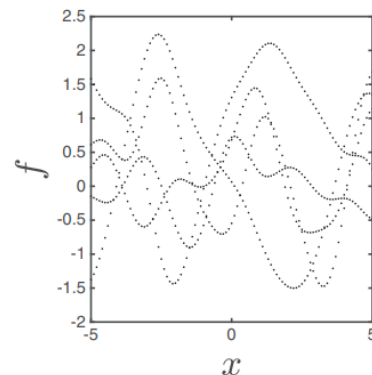
随着 x_n, x_m 距离增大， f_n, f_m 组成二维高斯分布的协方差矩阵不断趋于
对角矩阵



(a) $\gamma = 0.05$



(b) $\gamma = 0.1$



(c) $\gamma = 0.5$

- 直接从“先验预测分布”中进行采样，已经可以得到相互关联的样本(平滑的曲线)，但是，所采样得到的曲线是“随机的”，如何做出“合理的预测”？

结合有标签的“训样本点”，对曲线进行“准确的预测”

条件预测： $p(\mathbf{f}|\mathbf{f}^*, \mathbf{X})$

1. Gaussian Process (GP)-条件预测: $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X})$

- 高斯过程条件预测: $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X})$ 过程:

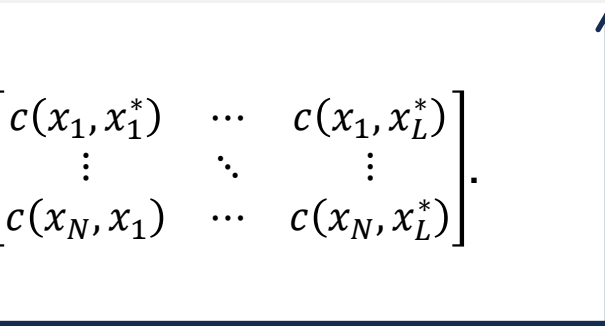
- 训练集: $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$, $\mathbf{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}$, 测试集: $\mathbf{x}^* = \begin{bmatrix} x_1^* \\ \vdots \\ x_L^* \end{bmatrix}$, $\mathbf{f}^* = \begin{bmatrix} f_1^* \\ \vdots \\ f_L^* \end{bmatrix}$.

- 先验预测分布: $p(\hat{\mathbf{f}}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{C} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{C}^* \end{bmatrix}\right)$, $\hat{\mathbf{f}} = \begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix}$.



其中, $\mathbf{C} = \begin{bmatrix} c(x_1, x_1) & \cdots & c(x_1, x_N) \\ \vdots & \ddots & \vdots \\ c(x_N, x_1) & \cdots & c(x_N, x_N) \end{bmatrix}$, $\mathbf{C}^* = \begin{bmatrix} c(x_1^*, x_1^*) & \cdots & c(x_1^*, x_L^*) \\ \vdots & \ddots & \vdots \\ c(x_L^*, x_1^*) & \cdots & c(x_L^*, x_L^*) \end{bmatrix}$, $\mathbf{R} = \begin{bmatrix} c(x_1, x_1^*) & \cdots & c(x_1, x_L^*) \\ \vdots & \ddots & \vdots \\ c(x_N, x_1^*) & \cdots & c(x_N, x_L^*) \end{bmatrix}$.

- 条件预测分布: $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X}) = \mathcal{N}(\mathbf{f}^*; \underline{\underline{\boldsymbol{\mu}^*}}, \underline{\underline{\boldsymbol{\Sigma}^*}})$. $\underline{\underline{\boldsymbol{\mu}^*}} = \mathbf{R}^\top \mathbf{C}^{-1} \mathbf{f}$, $\underline{\underline{\boldsymbol{\Sigma}^*}} = \mathbf{C}^* - \mathbf{R}^\top \mathbf{C}^{-1} \mathbf{R}$.



Comment 8.2 – Gaussian Conditionals: Consider a random vector $\mathbf{z}$ consisting of variables that can be split into two sets, $\mathbf{z}_1$ and $\mathbf{z}_2$:

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}.$$

Assume that $\mathbf{z}$ follows a Gaussian distribution:

$$p(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}.$$

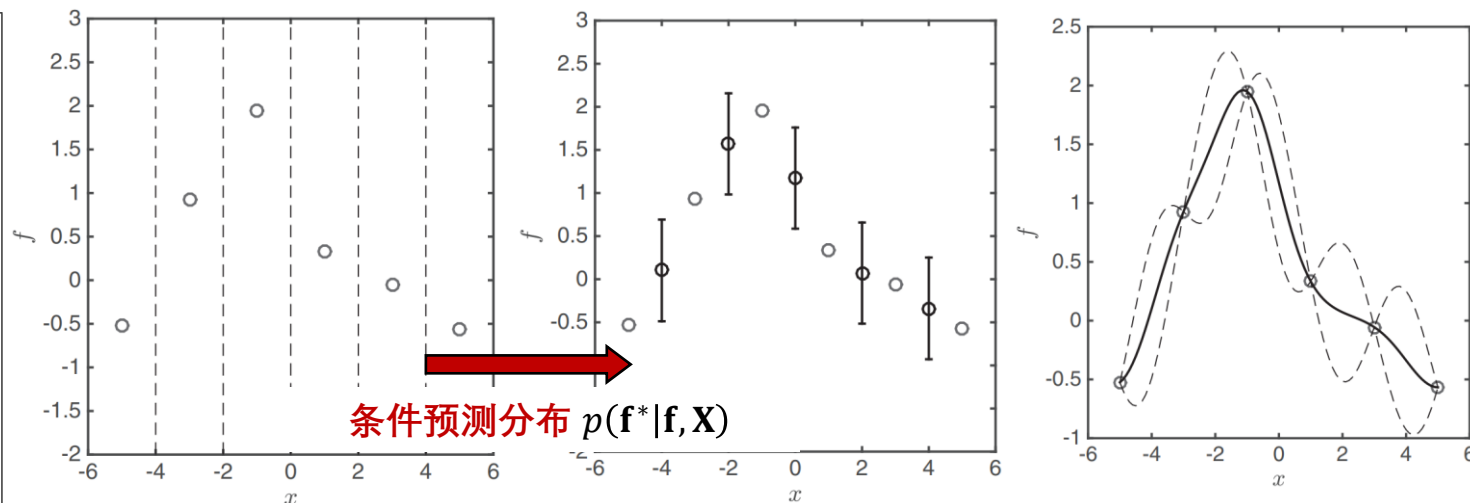
(Note that $\boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{12}^\top$ as covariance matrices must be symmetric.) If we observe the value of one of the sets of variables (say $\mathbf{z}_1$), the conditional density of the other set is also Gaussian:

$$p(\mathbf{z}_2|\mathbf{z}_1) = \mathcal{N}(\mathbf{a}, \mathbf{B})$$

$$\mathbf{a} = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{12}^\top \boldsymbol{\Sigma}_{11}^{-1}(\mathbf{z}_1 - \boldsymbol{\mu}_1), \quad \mathbf{B} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{12}^\top \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12}.$$

- 高斯过程 (GP) 存在的问题:

- 1. “核函数”- $c(x_n, x_m)$: 需要人为指定, 并且对于“条件预测分布”的影响较大;
- 2. “条件预测分布”: $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X}) = \mathcal{N}(\mathbf{f}^*; \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$. $\boldsymbol{\mu}^* = \mathbf{R}^\top \mathbf{C}^{-1} \mathbf{f}$, $\boldsymbol{\Sigma}^* = \mathbf{C}^* - \mathbf{R}^\top \mathbf{C}^{-1} \mathbf{R}$. 计算复杂度与训练、测试样本总数 $N + L$ 有关, 复杂度为 $\mathcal{O}((N + L)^3)$. 当样本总数较大时, 无法完成预测.



训练集: $\mathbf{x}, \mathbf{f}$

测试集: $\mathbf{x}^*, \mathbf{f}^*$

2. Conditional Neural Processes (CNP)

- 高斯过程 (GP) 存在的问题:

- 1. “核函数”- $c(x_n, x_m)$: 需要人为指定, 并且对于“条件预测分布”的影响较大;
- 2. “条件预测分布”: $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X}) = \mathcal{N}(\mathbf{f}^*; \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$. $\boldsymbol{\mu}^* = \mathbf{R}^\top \mathbf{C}^{-1} \mathbf{f}$, $\boldsymbol{\Sigma}^* = \mathbf{C}^* - \mathbf{R}^\top \mathbf{C}^{-1} \mathbf{R}$. 计算复杂度与训练、测试样本总数 $N + L$ 有关, 复杂度为 $\mathcal{O}((N + L)^3)$. 当样本总数较大时, 无法完成预测.

能否直接使用神经网络 θ 来描述 GP 中的“条件预测分布”: $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X})$?

- 条件神经过程 (CNP):

- 训练集 (Context Set): $\mathbf{C} = \{\mathbf{X}_C, \mathbf{Y}_C\} = \{(x^{(c)}, y^{(c)})\}_{c=1}^C$; 测试集 (Target Set): $\mathbf{T} = \{\mathbf{X}_T, \mathbf{Y}_T\} = \{(x^{(t)}, y^{(t)})\}_{t=1}^T$.
- “条件预测分布-似然” (T维高斯分布, 每个维度相互独立):

$$p_{\theta}(\mathbf{Y}_T|\mathbf{X}_T, \mathbf{C}) = \mathcal{N}(\mathbf{Y}_T|\boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T) = \prod_{t=1}^T p_{\theta}(y^{(t)}|x^{(t)}, \mathbf{C}) = \prod_{t=1}^T \mathcal{N}(y^{(t)}|\mu^{(t)}, \sigma^{2(t)}).$$

- Encoding (Enc_{θ}): $\mathbf{R} = \text{Enc}_{\theta}(\mathbf{C}) = \frac{1}{C} \sum_{c=1}^C \mathbf{R}^{(c)} = \frac{1}{C} \sum_{c=1}^C \text{Enc}_{\theta}((x^{(c)}, y^{(c)}))$.
- Decoding (Dec_{θ}): $(\mu^{(t)}, \sigma^{2(t)}) = \text{Dec}_{\theta}(\mathbf{R}, x^{(t)})$.

- 1. 与 GP 类似, 每个数据点的预测都依赖于 Context Set 的数据点
- 2. 假设每个样本 $y^{(t)}$ 都是“独立同分布的”, 忽略了 $\mathbf{X}_T$ 内样本之间的“位置关联性”

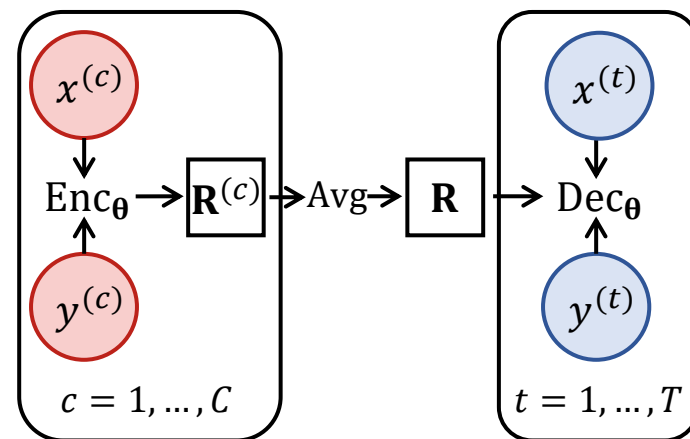
解决方法: CNP $\rightarrow$ Latent NP (LNP)

使用“平均”的目的: 1. 保证 $x^{(t)}$ 的预测“依赖于”所有 Context Set 内的样本点; (类似于 GP) 2. 保证预测结果与 $(x^{(c)}, y^{(c)})$ 的出现的先后顺序无关.

使用“平均”的问题: 所有 $(x^{(c)}, y^{(c)})$ 对于 $x^{(t)}$ 预测的“重要程度”相同, 但是, 直观上说, $(x^{(c)}, y^{(c)})$ 对于较近的 $x^{(t)}$ 的预测影响应该较大.

(在 GP 中, 可以通过“核函数”- $c(x^{(c)}, x^{(t)})$ 来动态描述不同样本间的“重要程度”)

解决方法: CNP $\rightarrow$ Attentive NP (AttnNP)



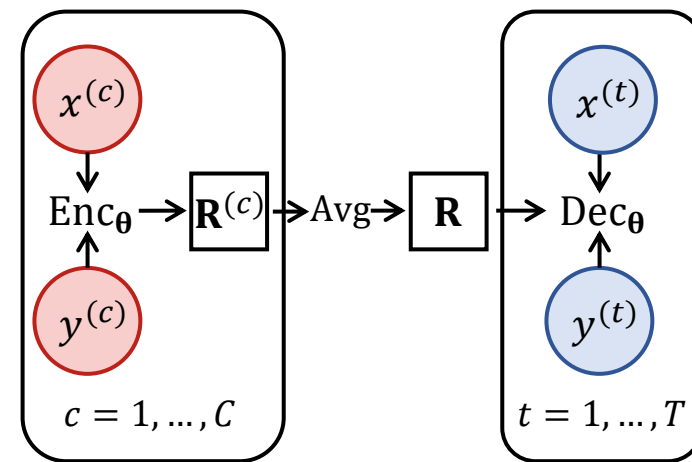
2. Conditional Neural Processes (CNP)-Training θ

● 条件神经过程 (CNP) :

- 训练集 (Context Set): $\mathbf{C} = \{\mathbf{X}_C, \mathbf{Y}_C\} = \{(x^{(c)}, y^{(c)})\}_{c=1}^C$; 测试集 (Target Set): $\mathbf{T} = \{\mathbf{X}_T, \mathbf{Y}_T\} = \{(x^{(t)}, y^{(t)})\}_{t=1}^T$.
- “条件预测分布”: $p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}) = \prod_{t=1}^T p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C}) = \prod_{t=1}^T \mathcal{N}(y^{(t)} | \mu^{(t)}, \sigma^{2(t)})$.
- Encoding (Enc_{θ}): $\mathbf{R} = \text{Enc}_{\theta}(\mathbf{C}) = \frac{1}{C} \sum_{c=1}^C R^{(c)} = \frac{1}{C} \sum_{c=1}^C \text{Enc}_{\theta}((x^{(c)}, y^{(c)}))$.
- Decoding (Dec_{θ}): $(\mu^{(t)}, \sigma^{2(t)}) = \text{Dec}_{\theta}(\mathbf{R}, x^{(t)})$.

如何获得好的 θ ?

即输入新的测试样本点 $x^{(t)}$, 能够获得正确的“条件预测分布 $p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C})$ ”



● CNP-训练过程:

- ① **Sample $Task_i = (\mathbf{C}, \mathbf{T}) \sim \mathcal{D}$** : 类似于元学习, 从整个数据集 $\mathcal{D}$ 内生成多个“Context Set: $\mathbf{C}$ ”和“Target Set: $\mathbf{T}$ ”, 构造多个“训练任务”;
- ② **$\mathcal{L}(\theta) = -\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C})$** : 该损失函数希望由 $\mathbf{X}_T, \mathbf{C}, \theta$ 所确定下的“条件预测分布”能够尽可能地符合观测数据 $\mathbf{Y}_T$. 即在该“条件预测分布”下的 $\mathbf{Y}_T$ 的似然尽可能大;
- ③ **$\theta^* \leftarrow \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta)$** ;
- ④ **重复①~③ N_{task} 次**: 目的是希望利用“多个不同的任务”, 训练 θ , 使其能够快速利用新任务的 Context Set, 准确地描述“条件预测分布”.

- **CNP-测试过程**: 给定新测试任务 $Task_{unseen}$ 的 Context Set, 利用 $\theta^*, \mathbf{X}_T$, 获得 $\mathbf{Y}_T$ 的“条件预测分布” $p_{\theta^*}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C})$.

2. Conditional Neural Processes (CNP)-Testing θ , Drawbacks

● CNP-测试过程:

- 给定新测试任务 $Task_{unseen}$ 的 Context Set, 利用 θ^* , X_T , 获得 Y_T 的“条件预测分布”:

$$p_{\theta^*}(Y_T|X_T, C) = \mathcal{N}(Y_T; \mu^*, \Sigma^*) = \prod_{t=1}^T \mathcal{N}(y^{(t)} | \mu^{(t)}, \sigma^{2(t)})$$

- Prediction Results** $\leftarrow \mu^*$. 将“条件预测分布”的均值向量 μ^* 作为预测结果.
- Prediction Uncertainty** $\leftarrow \Sigma^*$. 将“条件预测分布”的协方差矩阵(对角矩阵)的对角元素 $\Sigma_{t,t}^*$ 作为 $y^{(t)}$ 的预测方差.

由于Context Number较少, 因此, 直接取 μ^* 作为Target Set Y_T 的预测结果, 限制了模型的“不确定性预测能力”
(直观上说, 此时能够生成的图片应该不止一张, 并且“差异应该很大”)

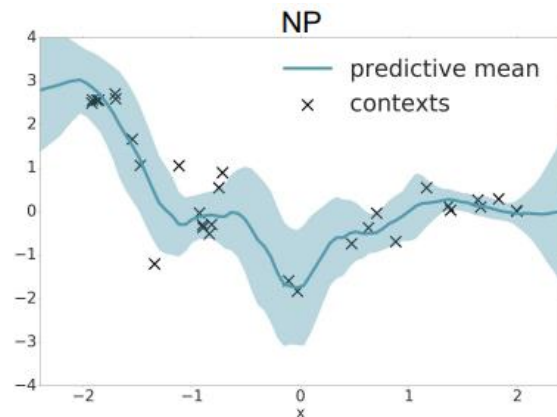
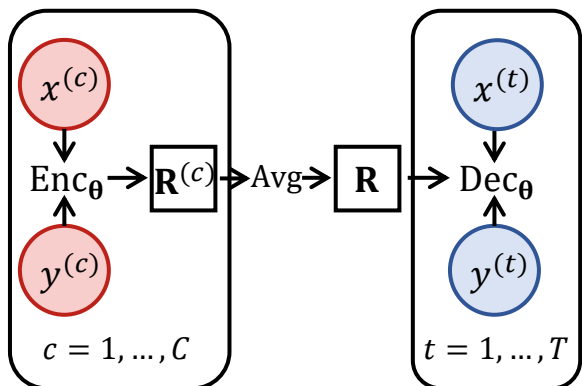


Fig. 1

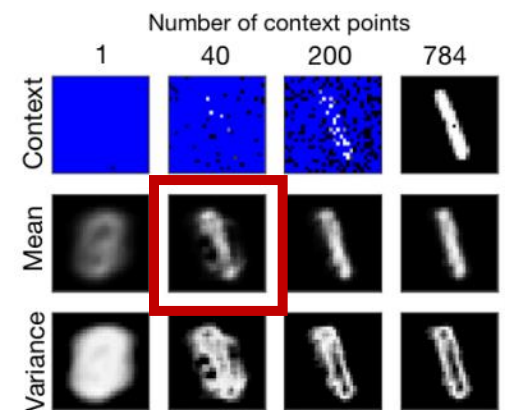


Fig. 2

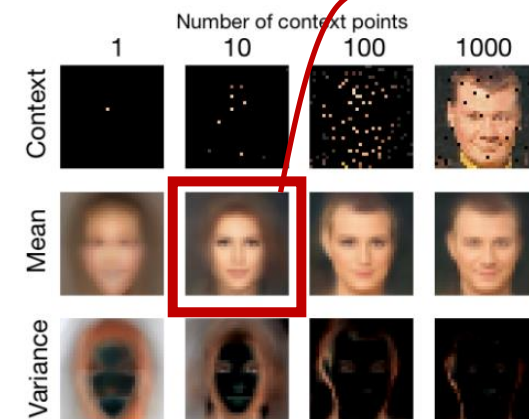


Fig. 3

● CNP-存在的问题:

- ① 直接使用各自的均值 $\mu^{(t)}$ 作为预测值, 限制了模型的不确定性预测能力. 未考虑和Target Set 内 $x^{(t)}$ 之间的“位置关系”. 例如Fig. 3. 但是, 简单地对每个高斯分布 $\mathcal{N}(y^{(t)} | \mu^{(t)}, \sigma^{2(t)})$ 进行采样 (因为我们假定每个预测标签 $y^{(t)}$ 都是“相互独立的”), 又会生成许多无关的预测结果 $y^{(1)} \dots y^{(T)}$, 这显然不合理, 因为相邻像素点 $y^{(t)}$ 和 $y^{(t+1)}$ 之间的采样应该“相互关联”. 如何保证 Y_T 内样本存在“相互关联”?
- ② 未考虑Context Set内样本点 $x^{(c)}$ 和Target Set 内 $x^{(t)}$ 的“位置关系”: Enc_{θ} 在使用C对新样本点 $x^{(t)}$ 进行预测时的“平均操作”, 认为所有 $(x^{(c)}, y^{(c)})$ 对于 $x^{(t)}$ 预测的“重要程度”相同, 但是, 直观上说, $(x^{(c)}, y^{(c)})$ 对于较近的 $x^{(t)}$ 的预测分布的影响应该较大.

1. Latent NP (LNP)

2. Attentive LNP (AttnLNP)

3. Latent Neural Processes (LNP)

- CNP-存在的问题：① 未考虑和Target Set 内 $x^{(t)}$ 之间的“位置关系”. 直接使用各自的均值 $\mu^{(t)}$ 作为预测值，限制了模型的预测能力

- **隐神经过程 (LNP)：** 引入隐变量 z , 并使用“边缘化条件预测分布”.

- 训练集 (Context Set): $\mathbf{C} = \{\mathbf{X}_C, \mathbf{Y}_C\} = \{(x^{(c)}, y^{(c)})\}_{c=1}^C$; 测试集 (Target Set): $\mathbf{T} = \{\mathbf{X}_T, \mathbf{Y}_T\} = \{(x^{(t)}, y^{(t)})\}_{t=1}^T$.

- “边缘化条件预测分布-边缘似然” (边缘化隐变量 z , 得到条件预测分布):

$$p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}) = \mathcal{N}(\mathbf{Y}_T | \boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T) = \int p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \mathbf{z}) p_{\theta}(\mathbf{z} | \mathbf{C}) d\mathbf{z}$$

$$= \int \prod_{t=1}^T p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C}, \mathbf{z}) p_{\theta}(\mathbf{z} | \mathbf{C}) d\mathbf{z} = \int \prod_{t=1}^T \mathcal{N}(y^{(t)} | \mu^{(t)}, \sigma^{2(t)}, \mathbf{z}) p_{\theta}(\mathbf{z} | \mathbf{C}) d\mathbf{z}$$

- Encoding (Enc_{θ}): $(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z) = \text{Enc}_{\theta}(\mathbf{C}) = \text{Enc}_{\theta}(\frac{1}{C} \sum_{c=1}^C R^{(c)})$
- Decoding (Dec_{θ}): $(\mu^{(t)}, \sigma^{2(t)}) = \sum_{m=1}^M \text{Dec}_{\theta}(x^{(t)}, \mathbf{z}_m)$

- 引入隐变量 z 使用“边缘似然”的好处:

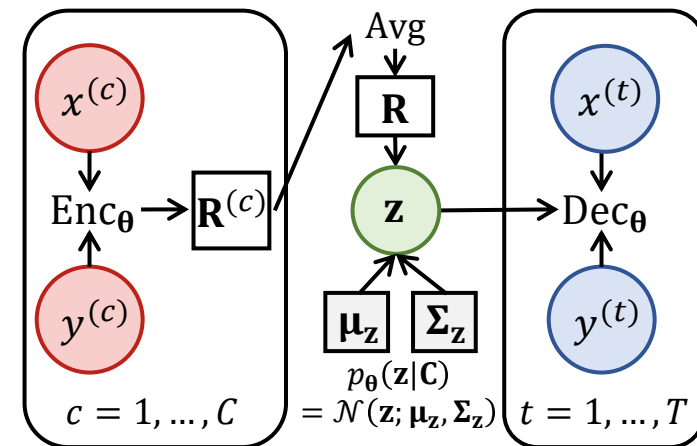
- 1. 可以根据采样得到的 z_i 生成“多组相互关联”的预测均值向量 μ_i .
- 2. 每个数据点 $y^{(t)}$ 的预测分布 $\int \mathcal{N}(y^{(t)} | \mu^{(t)}, \sigma^{2(t)}, \mathbf{z}) p_{\theta}(\mathbf{z} | \mathbf{C}) d\mathbf{z}$, 可以看作是“无限个高斯分布”相加(无限维高斯混合分布), 提高了预测模型的代表能力.

- **LNP需要解决的优化问题:**

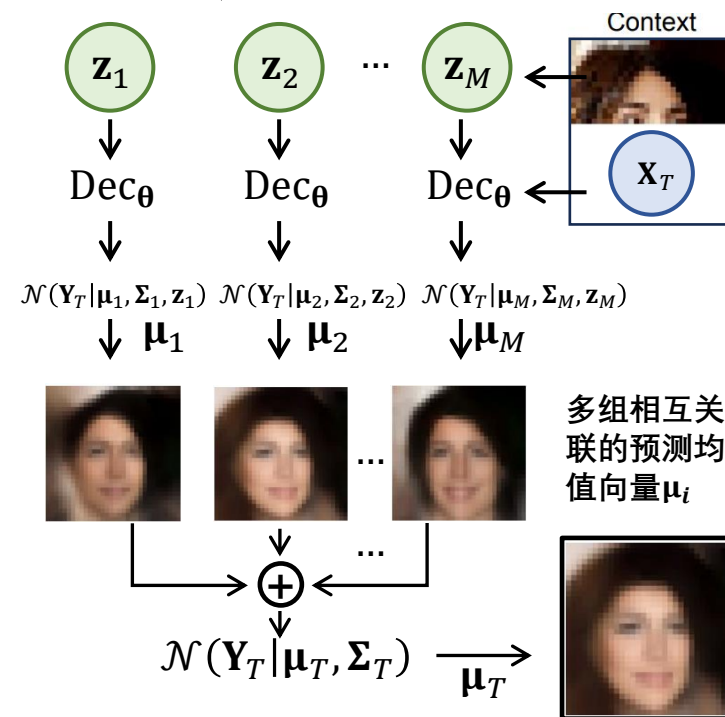
- 如何优化 θ , 找到合适的隐变量 z 的分布 $p_{\theta}(\mathbf{z} | \mathbf{C})$, 使其能够准确描述“边缘似然” $\int \mathcal{N}(\mathbf{Y}_T | \boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T, \mathbf{z}) p_{\theta}(\mathbf{z} | \mathbf{C}) d\mathbf{z}$?

1. Maximum Likelihood-LNP (ML-LNP)

2. Variational Inference-LNP (VI-LNP)



LNP预测过程



3.1 Latent Neural Processes (LNP) Training θ : Maximum Likelihood-LNP (ML-LNP)

● 隐神经过程 (LNP) :

- 训练集 (Context Set): $\mathbf{C} = \{\mathbf{X}_C, \mathbf{Y}_C\} = \{(x^{(c)}, y^{(c)})\}_{c=1}^C$; 测试集 (Target Set): $\mathbf{T} = \{\mathbf{X}_T, \mathbf{Y}_T\} = \{(x^{(t)}, y^{(t)})\}_{t=1}^T$.
- “边缘化条件预测分布-边缘似然” (边缘化隐变量 $\mathbf{z}$, 得到条件预测分布):

$$p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}) = \int p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \mathbf{z}) p_{\theta}(\mathbf{z} | \mathbf{C}) d\mathbf{z} = \int \prod_{t=1}^T p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C}, \mathbf{z}) p_{\theta}(\mathbf{z} | \mathbf{C}) d\mathbf{z} = \int \prod_{t=1}^T \mathcal{N}(y^{(t)} | \mu^{(t)}, \sigma^{2(t)}, \mathbf{z}) p_{\theta}(\mathbf{z} | \mathbf{C}) d\mathbf{z}$$

如何获得好的 θ ?

ML-LNP: 考虑直接极大化上述“边缘化条件预测分布”

$$\begin{aligned} \log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}) &\approx \log \left(\frac{1}{M} \left(\sum_{m=1}^M \prod_{t=1}^T p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C}, \mathbf{z}_m) \right) \right) \text{ (Monte Carlo 估计, } \mathbf{z}_m \sim p_{\theta}(\mathbf{z} | \mathbf{C}) \text{)} \\ &= \log \left(\sum_{m=1}^M \exp \left(\sum_{t=1}^T \log p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C}, \mathbf{z}_m) \right) \right) - \log M \\ &= \text{LogSumExp}_{m=1}^M \left[\sum_{t=1}^T \log p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C}, \mathbf{z}_m) \right] - \log M \text{ (代入LogSumExp函数计算)} \end{aligned}$$

● ML-LNP -训练过程:

- ① Sample $\text{Task}_i = (\mathbf{C}, \mathbf{T}) \sim \mathcal{D}$;
 - ② $\mathcal{L}(\theta) = -\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}) \approx \log M - \text{LogSumExp}_{m=1}^M \sum_{t=1}^T \log p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C}, \mathbf{z}_m)$. 极小化“负对数边缘似然”;
 - ③ $\theta^* \leftarrow \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta)$;
 - ④ 重复①~③ N_{task} 次;
- 此处涉及到Monte Carlo采样, 需要利用“重参数化技巧” (例如3.2 VI-LNP)

● ML-LNP的缺点:

- 直接对“边缘似然”的Monte Carlo 估计是从Context Set确定的隐变量分布 $p_{\theta}(\mathbf{z} | \mathbf{C})$ 中进行采样的, 可以视作是从“先验分布”中进行采样, 方差往往较大, 采到很多对边缘似然没有贡献的样本, 也就是说, 采样得到的 $\mathbf{z}_m$ 无法使似然 $p_{\theta}(y^{(t)} | x^{(t)}, \mathbf{C}, \mathbf{z}_m)$ 较大.
- **VI-LNP:** 使用变分推断方法, 从近似后验分布 $q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T})$ 中进行采样, 目的是采样更多对训练有用的样本 $\mathbf{z}_m$.

3.2 Latent Neural Processes (LNP)-Training θ : Variational Inference-LNP (VI-LNP)

- ML-LNP: $\theta^* \leftarrow \underset{\theta}{\operatorname{argmax}} p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C})$

使用变分推断方法, 从近似后验分布 $q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T})$ 中进行采样, 目的是获得更多对训练有用的样本 $\mathbf{z}_m$.

$$\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}) \geq \text{ELBO}(\theta) = \mathbb{E}_{q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T})} [\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \mathbf{z})] - \text{KL}(q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T}) || p_{\theta}(\mathbf{z} | \mathbf{C}))$$

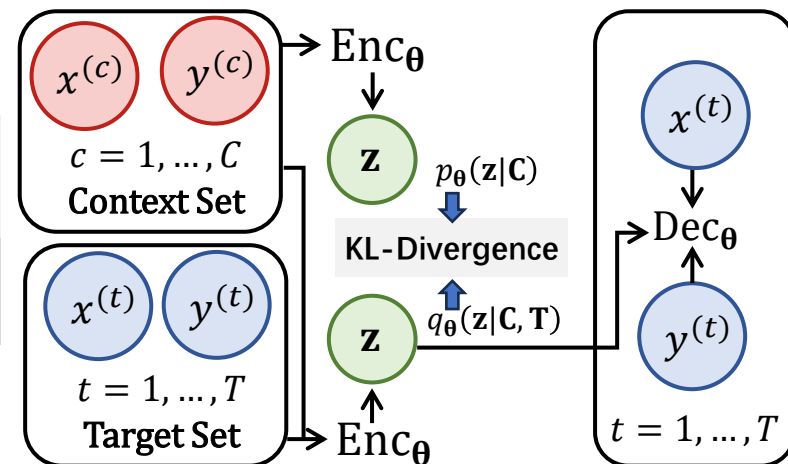
- VI-LNP: $\theta^* \leftarrow \underset{\theta}{\operatorname{argmax}} \text{ELBO}(\theta)$

- 明确ELBO内几个变量的具体形式:

- 1. 拟合似然项: $\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \mathbf{z}) = \prod_{t=1}^T \mathcal{N}(y^{(t)} | \mu^{(t)}, \sigma^{2(t)}, \mathbf{z})$. 从变分分布 $q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T})$ 中采样 $\mathbf{z}_m$, 结合 $x^{(t)}$, 确定预测高斯分布.
- 2. 变分分布: $q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T}) = \text{Enc}_{\theta}(\mathbf{C}, \mathbf{T})$. 使用Context Set和Target Set内的所有样本点, 计算“近似后验分布”的“均值”和“方差”.
- 3. 先验分布: $p_{\theta}(\mathbf{z} | \mathbf{C}) = \text{Enc}_{\theta}(\mathbf{C})$. 使用Context Set内的样本点, 计算“先验分布”的“均值”和“方差”.

- 基于梯度优化ELBO(θ): 需要对 θ 计算梯度, 即 $\nabla_{\theta} \text{ELBO}(\theta)$

$$\begin{aligned} \nabla_{\theta} \text{ELBO}(\theta) &= \nabla_{\theta} \{ \mathbb{E}_{q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T})} [\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \mathbf{z})] - \text{KL}(q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T}) || p_{\theta}(\mathbf{z} | \mathbf{C})) \} \\ &= \nabla_{\theta} \mathbb{E}_{q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T})} [\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \mathbf{z})] \text{ (期望的梯度)} - \nabla_{\theta} \text{KL}(q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T}) || p_{\theta}(\mathbf{z} | \mathbf{C})) \\ &= \nabla_{\theta} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} [\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \epsilon)] \text{ (重参数化技巧)} - \nabla_{\theta} \text{KL}(q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T}) || p_{\theta}(\mathbf{z} | \mathbf{C})) \\ &= \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} [\nabla_{\theta} \log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \epsilon)] \text{ (梯度的期望)} - \nabla_{\theta} \text{KL}(q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T}) || p_{\theta}(\mathbf{z} | \mathbf{C})) \\ &\approx \frac{1}{M} \sum_{m=1}^M [\nabla_{\theta} \log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \epsilon_m)] \text{ (Monte Carlo 估计)} - \nabla_{\theta} \text{KL}(q_{\theta}(\mathbf{z} | \mathbf{C}, \mathbf{T}) || p_{\theta}(\mathbf{z} | \mathbf{C})) \end{aligned}$$



VI-LNP 训练阶段前向传播过程

直接将Context Set和Target Set经过 Enc_{θ} 输出的“均值”和“方差”代入KL散度计算公式, 并计算梯度

- VI-LNP -训练过程: 与ML-LNP基本一致, 只是第②步中的损失函数变为 $\mathcal{L}(\theta) = -\text{ELBO}(\theta)$.

3.2 Latent Neural Processes (LNP)-Testing θ : Variational Inference-LNP (VI-LNP)

- VI-LNP: $\theta^* \leftarrow \underset{\theta}{\operatorname{argmax}} \operatorname{ELBO}(\theta)$.

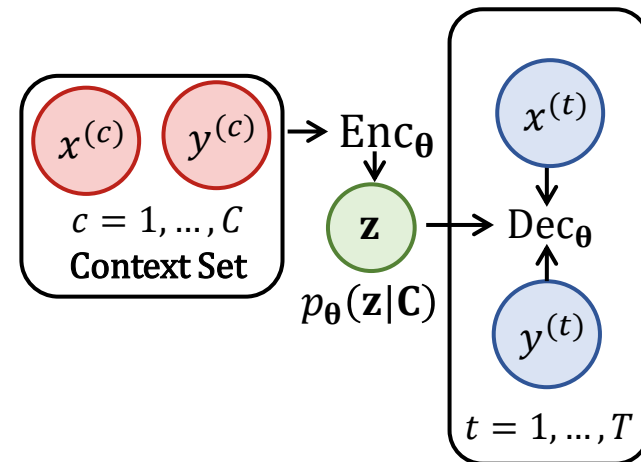
$$\operatorname{ELBO}(\theta) = \mathbb{E}_{q_{\theta}(\mathbf{z}|\mathbf{C}, \mathbf{T})} [\log p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}, \mathbf{z})] - \underbrace{\operatorname{KL}(q_{\theta}(\mathbf{z}|\mathbf{C}, \mathbf{T}) || p_{\theta}(\mathbf{z}|\mathbf{C}))}_{\text{KL Divergence}}$$

● VI-LNP –测试过程:

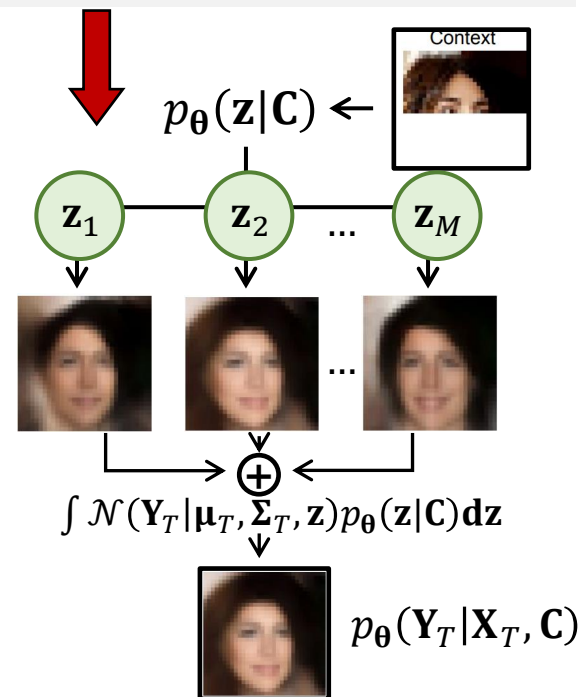
- ① 给定新测试任务 $Task_{\text{unseen}}$ 的 Context Set: $\mathbf{C} = \{\mathbf{X}_C, \mathbf{Y}_C\} = \{(x^{(c)}, y^{(c)})\}_{c=1}^C$. Target Set: $\mathbf{T} = \{\mathbf{X}_T\} = \{x^{(t)}\}_{t=1}^T$.
- ② 根据 Context Set 获得隐变量 $\mathbf{z}$ 的“先验分布 $p_{\theta}(\mathbf{z}|\mathbf{C})$ ”.
注意, 因为测试任务的标签 $\mathbf{Y}_T$ 并没有给出, 因此我们不能像训练过程一样获得 $\mathbf{z}$ 的近似后验分布 $q_{\theta}(\mathbf{z}|\mathbf{C}, \mathbf{T})$. 但是, 根据 $\operatorname{ELBO}(\theta)$ 内的 $\operatorname{KL}(q_{\theta}(\mathbf{z}|\mathbf{C}, \mathbf{T}) || p_{\theta}(\mathbf{z}|\mathbf{C}))$ 项, 实际上已经迫使模型在新的任务上, 仅利用 Context Set, 就能够快速地生成与近似后验 $q_{\theta}(\mathbf{z}|\mathbf{C}, \mathbf{T})$ 一致的先验 $p_{\theta}(\mathbf{z}|\mathbf{C})$.
- ③ 测试类型一: 利用 $p_{\theta}(\mathbf{z}|\mathbf{C})$, 计算“边缘似然” $p_{\theta}(\mathbf{Y}_T | \mathbf{X}_T, \mathbf{C}) = \int \mathcal{N}(\mathbf{Y}_T | \boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T, \mathbf{z}) p_{\theta}(\mathbf{z}|\mathbf{C}) d\mathbf{z}$.
- ③ 测试类型二: 从 $p_{\theta}(\mathbf{z}|\mathbf{C})$ 中采样多个 $\mathbf{z}_m$, 生成多个相关联的预测样本.

● VI-LNP的缺点:

- 仍未考虑 Context Set 和 Target Set 内样本点的位置关系: 由于在 $\operatorname{Enc}_{\theta}(\mathbf{C})$ 前向传播的过程中, 对于所有 Context Set 内样本点的特征 $R^{(c)}$ 使用了“平均操作” $\operatorname{Enc}_{\theta}(\frac{1}{C} \sum_{c=1}^C R^{(c)})$. 因此, 仍然无法做到类似 Gaussian Process 中预测条件分布所描述的“ $(x^{(c)}, y^{(c)})$ 对于较近的 $x^{(t)}$ 的预测影响应该较大”.

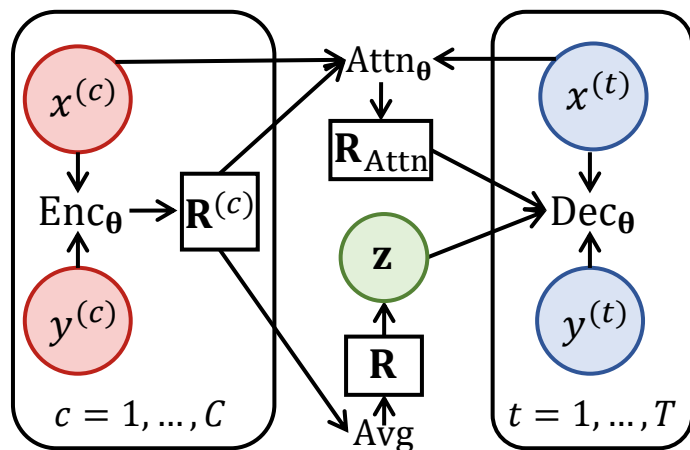


VI-LNP 测试阶段前向传播过程

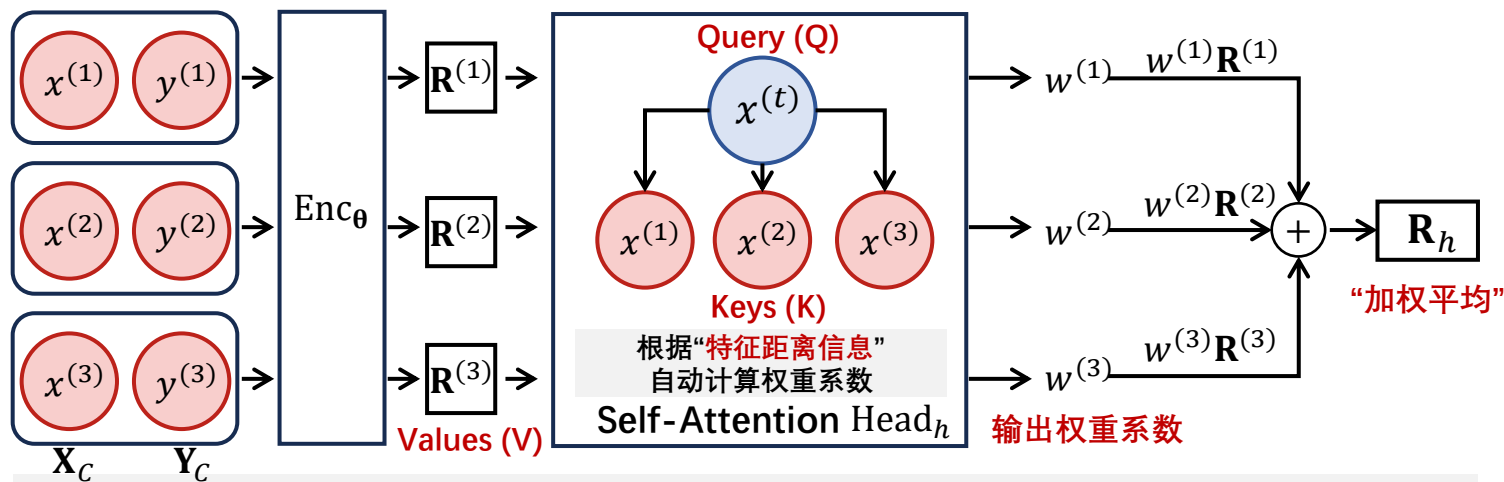


LNP → Attentive LNP (AttnLNP)

4. Attentive LNP (AttnLNP)-MultiHead-Self-Attention



AttnLNP预测过程



自注意力头Head_h前向传播过程: $R_h = \text{DotProduct}(QW_h^Q, KW_h^K, VW_h^V)$

- VI-LNP的缺点：仍未考虑Context Set和Target Set内样本点的位置关系。

- **AttnLNP**：利用 X_C 和 X_T 之间的“距离信息”对特征 $R^{(c)}$ 进行“加权平均”。

• 训练集 (Context Set): $C = \{X_C, Y_C\} = \{(x^{(c)}, y^{(c)})\}_{c=1}^C$;

测试集 (Target Set): $T = \{X_T, Y_T\} = \{(x^{(t)}, y^{(t)})\}_{t=1}^T$ 。

• “边缘似然”: $p_\theta(Y_T|X_T, C) = \mathcal{N}(Y_T|\mu_T, \Sigma_T) = \int \mathcal{N}(Y_T|\mu_T, \Sigma_T, z)p_\theta(z|C)dz$ 。

• Encoding (Enc_θ): $R = \text{Enc}_\theta(C) = \frac{1}{C} \sum_{c=1}^C R^{(c)}$; $R^{(c)} = \text{Enc}_\theta((x^{(c)}, y^{(c)}))$ 。

• **Attention (Attn_θ)**: $R_{\text{Attn}} = \text{Concat}(R_1, R_2, \dots, R_H)W$, $R_h = \text{DotProduct}(QW_h^Q, KW_h^K, VW_h^V)$ 。

(实践中, 使用MultiHead-Self-Attention $\text{Head}_{h=1, \dots, H}$; 注意, 在AttnLNP中不对生成隐变量z的R计算自注意力系数, 进行加权平均, 是因为在LNP中, 认为隐变量z描述的是 X_T 内样本之间的位置关系, 而不是 X_C 与 X_T 之间的位置关系。)

• Decoding (Dec_θ): $(\mu^{(t)}, \sigma^{2(t)}) = \sum_{m=1}^M \text{Dec}_\theta(x^{(t)}, z_m, R_{\text{Attn}})$

- AttnLNP 的训练, 测试过程: 与LNP基本一致, 只是前向传播过程中需要额外计算 R_{Attn} 。

在自注意力头(Head_h)中:

- ① **Keys (K)**: X_C ; 将Context Set的特征 X_C 视为“位置信息”,
- ② **Query (Q)**: $x^{(t)}$;
- ③ **Values (V)**: $R^{(c)}$;
- ④使用 W_h^Q, W_h^K, W_h^V , 自动计算 $R^{(c)}$ 的权重系数 $w^{(c)}$;
- ⑤ “加权平均”, 得到 R_h 。