Scalable and Deep Gaussian Processes

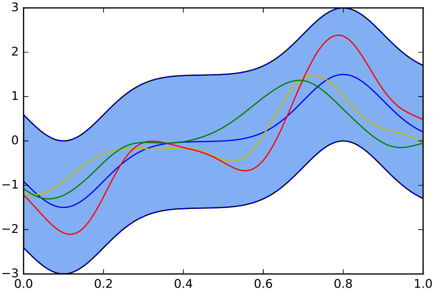
Dmitry A. Kropotov

DeepBayes

Gaussian Process

Gaussian Process (GP) is a stochastic process over real-valued functions.

$$
\begin{aligned}
& f(\boldsymbol{x}) \sim GP(m(\cdot), k(\cdot,\cdot)) \;\Leftrightarrow \\
& \boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)] \sim \mathcal{N}(\boldsymbol{f} | \boldsymbol{m}_n, K_{nn}), \\
& \boldsymbol{m}_n = [m(\boldsymbol{x}_1), \ldots, m(\boldsymbol{x}_n)], \\
& K_{nn} = \{k(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i,j=1}^{n,n}.
\end{aligned}
$$



Gaussian Process for regression and classification

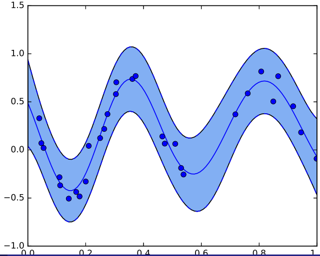
Suppose we have a dataset $(\boldsymbol{y}, X) = \{y_i, \boldsymbol{x}_i\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathbb{R}^d$ – feature vectors, y_i – target values.

$$p(\boldsymbol{y}, \boldsymbol{f} | X, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}|X, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(y_i|f_i) p(\boldsymbol{f}|X, \boldsymbol{\theta}),$$

$$p(\boldsymbol{f}|X, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, K(X, X; \boldsymbol{\theta})) \text{ – GP},$$

$$p(y_i|f_i) = \mathcal{N}(y_i|f_i, \sigma^2) \text{ – regression},$$

$$p(y_i|f_i) = \frac{1}{1 + \exp(-y_i f_i)} \text{ – classification with 2 classes.}$$



GP probabilistic model:

$$p(\boldsymbol{y}, \boldsymbol{f}|X, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|X, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(y_i|f_i)p(\boldsymbol{f}|X, \boldsymbol{\theta}).$$

Training:

$$\log p(\boldsymbol{y}|X, \boldsymbol{\theta}) \geq \mathbb{E}_{q(\boldsymbol{f})}[\log p(\boldsymbol{y}, \boldsymbol{f}|X, \boldsymbol{\theta}) - \log q(\boldsymbol{f})] \to \max_{\boldsymbol{\theta}, q(\cdot)},$$
$$q(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \Sigma) \approx p(\boldsymbol{f}|\boldsymbol{y}, X, \boldsymbol{\theta}).$$

Testing:

$$p(\boldsymbol{f}_{test}|\boldsymbol{y}, \boldsymbol{\theta}, X, X_{test}) = \int p(\boldsymbol{f}_{test}|\boldsymbol{f}, \boldsymbol{\theta}, X, X_{test})p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{\theta}, X)d\boldsymbol{f} \approx$$
$$\approx \int p(\boldsymbol{f}_{test}|\boldsymbol{f}, \boldsymbol{\theta}, X, X_{test})q(\boldsymbol{f})d\boldsymbol{f}.$$

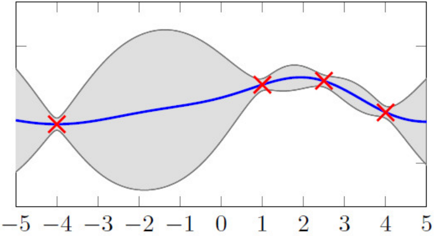
GP properties

Pros:

- Automatically adjust all parameters during training;
- Provides variance during prediction;

Cons:

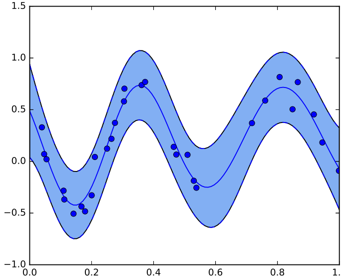
- Training scales as $O(n^3)$, where n – training sample size;
- Do not allow very complex dependencies like deep neural nets.

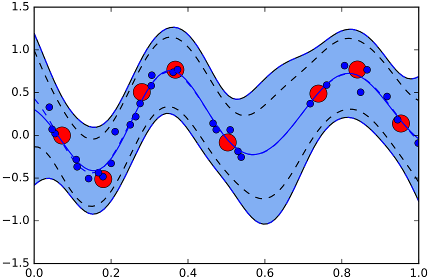


Gaussian Process with inducing inputs

Let's introduce some new points $Z = \{\boldsymbol{z}_i\}_{i=1}^m$ and suppose we know GP values at these points $\boldsymbol{u}$. Key assumption:

$$p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{u}, X, Z) \approx p(\boldsymbol{f}|\boldsymbol{u}, X, Z).$$





Gaussian Process with inducing inputs

Augmented model:

$$p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta}),$$
$$p(\boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta}) = \mathcal{N}([\boldsymbol{f}, \boldsymbol{u}]|[\mathbf{0}_n, \mathbf{0}_m], K_{n+m,n+m}).$$

Augmented model coincides with previous one:

$$\int p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta})d\boldsymbol{u} = p(\boldsymbol{y}, \boldsymbol{f}|X, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{f})\mathcal{N}(\boldsymbol{f}|\mathbf{0}_n, K_{nn}).$$

Augmented model:

$$p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u} | X, Z, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}, \boldsymbol{u} | X, Z, \boldsymbol{\theta}),$$
$$p(\boldsymbol{f}, \boldsymbol{u} | X, Z, \boldsymbol{\theta}) = \mathcal{N}([\boldsymbol{f}, \boldsymbol{u}] | [\mathbf{0}_n, \mathbf{0}_m], K_{n+m,n+m}).$$

Augmented model coincides with previous one:

$$\int p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u} | X, Z, \boldsymbol{\theta}) d\boldsymbol{u} = p(\boldsymbol{y}, \boldsymbol{f} | X, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{f}) \mathcal{N}(\boldsymbol{f} | \mathbf{0}_n, K_{nn}).$$

Training:

$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \mathbb{E}_{q(\boldsymbol{f}, \boldsymbol{u})} [\log p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u} | X, Z, \boldsymbol{\theta}) - \log q(\boldsymbol{f}, \boldsymbol{u})] \to \max_{\boldsymbol{\theta}, q(\boldsymbol{f}, \boldsymbol{u})},$$
$$q(\boldsymbol{f}, \boldsymbol{u}) \approx p(\boldsymbol{f}, \boldsymbol{u} | \boldsymbol{y}, X, Z, \boldsymbol{\theta}) = p(\boldsymbol{f} | \boldsymbol{u}, \boldsymbol{y}, X, Z, \boldsymbol{\theta}) p(\boldsymbol{u} | \boldsymbol{y}, X, Z, \boldsymbol{\theta}).$$

Augmented model:

$$p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta}),$$
$$p(\boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta}) = \mathcal{N}([\boldsymbol{f}, \boldsymbol{u}]|[\mathbf{0}_n, \mathbf{0}_m], K_{n+m,n+m}).$$

Augmented model coincides with previous one:

$$\int p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta})d\boldsymbol{u} = p(\boldsymbol{y}, \boldsymbol{f}|X, \boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{f})\mathcal{N}(\boldsymbol{f}|\mathbf{0}_n, K_{nn}).$$

Training:

$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}[\log p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta}) - \log q(\boldsymbol{f}, \boldsymbol{u})] \to \max_{\boldsymbol{\theta}, q(\boldsymbol{f},\boldsymbol{u})},$$

$$q(\boldsymbol{f}, \boldsymbol{u}) \approx p(\boldsymbol{f}, \boldsymbol{u}|\boldsymbol{y}, X, Z, \boldsymbol{\theta}) = p(\boldsymbol{f}|\boldsymbol{u}, \boldsymbol{y}, X, Z, \boldsymbol{\theta})p(\boldsymbol{u}|\boldsymbol{y}, X, Z, \boldsymbol{\theta}).$$

Using key assumption we can choose $q(\boldsymbol{f}, \boldsymbol{u})$ as follows:

$$q(\boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{f}|\boldsymbol{u}, X, Z, \boldsymbol{\theta})q(\boldsymbol{u}) = p(\boldsymbol{f}|\boldsymbol{u}, X, Z, \boldsymbol{\theta})\mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \Sigma).$$

Family for approximate posterior:

$$q(\boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{f}|\boldsymbol{u}, X, Z, \boldsymbol{\theta})q(\boldsymbol{u}).$$

Training:

$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})} \log \frac{p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{u}|X, Z, \boldsymbol{\theta})}{q(\boldsymbol{f}, \boldsymbol{u})} =$$

$$= \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})} \log \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{u}, X, Z, \boldsymbol{\theta})p(\boldsymbol{u}|Z, \boldsymbol{\theta})}{p(\boldsymbol{f}|\boldsymbol{u}, X, Z, \boldsymbol{\theta})q(\boldsymbol{u})} =$$

$$= \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})} \log p(\boldsymbol{y}|\boldsymbol{f}) + \mathbb{E}_{q(\boldsymbol{u})} \log \frac{p(\boldsymbol{u}|Z, \boldsymbol{\theta})}{q(\boldsymbol{u})} =$$

$$= \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \log p(y_i|f_i) - \mathrm{KL}(q(\boldsymbol{u})\|p(\boldsymbol{u}|Z, \boldsymbol{\theta})).$$

Gaussian Process with inducing inputs

Optimization criterion:

$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \log p(y_i|f_i) - \mathrm{KL}(q(\boldsymbol{u}) \| p(\boldsymbol{u}|Z, \boldsymbol{\theta})) \to \max_{\boldsymbol{\theta}, \boldsymbol{\mu}, \Sigma}.$$

First term:

$$q(\boldsymbol{f}) = \int q(\boldsymbol{f}, \boldsymbol{u}) d\boldsymbol{u} = \int p(\boldsymbol{f}|\boldsymbol{u}, X, Z) q(\boldsymbol{u}) d\boldsymbol{u} =$$

$$= \int \mathcal{N}(\boldsymbol{f}|K_{nm}K_{mm}^{-1}\boldsymbol{u}, K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}) \mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \Sigma) d\boldsymbol{u} =$$

$$= \mathcal{N}(\boldsymbol{f}|K_{nm}K_{mm}^{-1}\boldsymbol{\mu}, K_{nn} + K_{nm}K_{mm}^{-1}(\Sigma - K_{mm})K_{mm}^{-1}K_{mn}).$$

Optimization criterion:

$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \log p(y_i|f_i) - \mathrm{KL}(q(\boldsymbol{u}) \| p(\boldsymbol{u}|Z, \boldsymbol{\theta})) \to \max_{\boldsymbol{\theta}, \boldsymbol{\mu}, \Sigma}.$$

First term:

$$q(\boldsymbol{f}) = \int q(\boldsymbol{f}, \boldsymbol{u}) d\boldsymbol{u} = \int p(\boldsymbol{f}|\boldsymbol{u}, X, Z) q(\boldsymbol{u}) d\boldsymbol{u} =$$
$$= \int \mathcal{N}(\boldsymbol{f}|K_{nm}K_{mm}^{-1}\boldsymbol{u}, K_{nn} - K_{nm}K_{mm}^{-1}K_{mn})\mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \Sigma) d\boldsymbol{u} =$$
$$= \mathcal{N}(\boldsymbol{f}|K_{nm}K_{mm}^{-1}\boldsymbol{\mu}, K_{nn} + K_{nm}K_{mm}^{-1}(\Sigma - K_{mm})K_{mm}^{-1}K_{mn}).$$

$$q(f_i) = \mathcal{N}(f_i|\boldsymbol{k}_i^T K_{mm}^{-1}\boldsymbol{\mu}, k_{ii} + \boldsymbol{k}_i^T K_{mm}^{-1}(\Sigma - K_{mm})K_{mm}^{-1}\boldsymbol{k}_i),$$
$$\boldsymbol{k}_i = \{k(\boldsymbol{x}_i, \boldsymbol{z}_j)\}_{j=1}^{m}; \quad k_{ii} = k(\boldsymbol{x}_i, \boldsymbol{x}_i).$$

We don't need to work with K_{nn}, only with its diagonal!

Optimization criterion:

$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \log p(y_i|f_i) - \mathrm{KL}(q(\boldsymbol{u})\|p(\boldsymbol{u}|Z, \boldsymbol{\theta})) \to \max_{\boldsymbol{\theta}, \boldsymbol{\mu}, \Sigma}.$$

Second term:

$$\mathrm{KL}(q(\boldsymbol{u}\|p(\boldsymbol{u}|Z, \boldsymbol{\theta})) = \mathrm{KL}(\mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \Sigma)\|\mathcal{N}(\boldsymbol{u}|\mathbf{0}_m, K_{mm})) =$$
$$= -\frac{m}{2} - \frac{1}{2}\log\det K_{mm}^{-1}\Sigma + \frac{1}{2}\mathrm{tr} K_{mm}^{-1}\Sigma + \frac{1}{2}\boldsymbol{\mu}^T K_{mm}^{-1}\boldsymbol{\mu}.$$

Total costs for all $q(f_i)$ and the second term:

Optimization criterion:

$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \log p(y_i|f_i) - \mathrm{KL}(q(\boldsymbol{u}) \| p(\boldsymbol{u}|Z, \boldsymbol{\theta})) \to \max_{\boldsymbol{\theta}, \boldsymbol{\mu}, \Sigma}.$$

Second term:

$$\begin{aligned}\mathrm{KL}(q(\boldsymbol{u} \| p(\boldsymbol{u}|Z, \boldsymbol{\theta})) &= \mathrm{KL}(\mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \Sigma) \| \mathcal{N}(\boldsymbol{u}|\mathbf{0}_m, K_{mm})) = \\ &= -\frac{m}{2} - \frac{1}{2} \log\det K_{mm}^{-1}\Sigma + \frac{1}{2} \mathrm{tr} K_{mm}^{-1}\Sigma + \frac{1}{2} \boldsymbol{\mu}^T K_{mm}^{-1} \boldsymbol{\mu}.\end{aligned}$$

Total costs for all $q(f_i)$ and the second term:

$$O(nm^2 + m^3).$$

Computational issues

We need to estimate $\mathbb{E}_{q(f_i)} \log p(y_i|f_i)$.

In case of regression:

$$p(y_i|f_i) = \mathcal{N}(y_i|f_i, \sigma^2),$$
$$\mathbb{E}_{\mathcal{N}(f_i|m_i,s_i^2)} \log p(y_i|f_i) = \log \mathcal{N}(f_i|m_i, \sigma^2) - \frac{1}{2}s_i^2.$$

In case of classification:

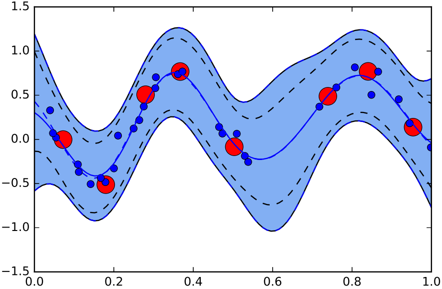
$$p(y_i|f_i) = \frac{1}{1 + \exp(-y_i f_i)},$$
$$\mathbb{E}_{\mathcal{N}(f_i|m_i,s_i^2)} \log p(y_i|f_i) = \mathbb{E}_{\mathcal{N}(\xi|0,1)} \log p(y_i|s_i\xi + m_i).$$

Expectation w.r.t. standard normal distribution can be estimated using Gauss-Hermite quadrature.

- Stochastic optimization with one epoch complexity $O(nm^2 + m^3)$;
- Can optimize w.r.t. positions of inducing points Z, but usually take them fixed, e.g. as cluster centres.

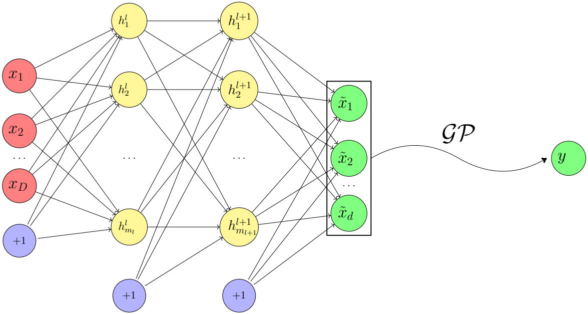
Limitations:

- Can't handle many inducing inputs;
- Current model is not deep.



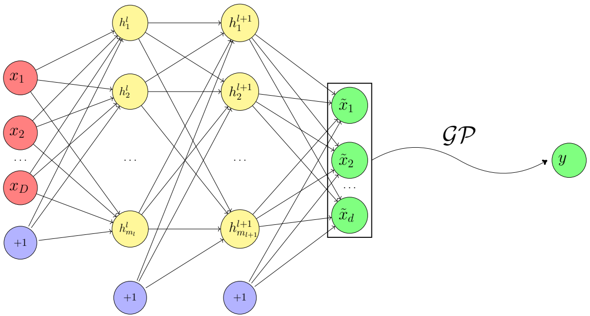
Use new covariance function:

$$k(\boldsymbol{x}, \boldsymbol{y}) = k(\mathrm{net}(\boldsymbol{x}; \boldsymbol{\eta}), \mathrm{net}(\boldsymbol{y}; \boldsymbol{\eta})).$$



Use new covariance function:

$$k(\boldsymbol{x}, \boldsymbol{y}) = k(\mathrm{net}(\boldsymbol{x}; \boldsymbol{\eta}), \mathrm{net}(\boldsymbol{y}; \boldsymbol{\eta})).$$



It is not clear where to put inducing inputs!

For two matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{p\times q}$ their Kronecker product is $np\times mq$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1m}B \\ \dots & \ddots & \dots \\ a_{n1}B & \dots & a_{nm}B \end{bmatrix}.$$

Properties:

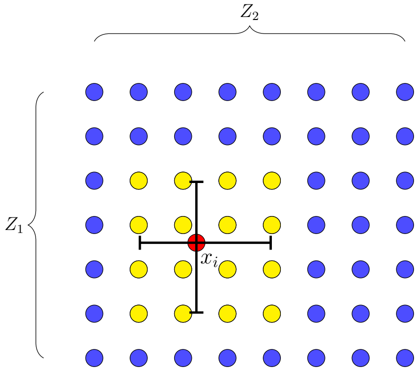
- $(A_1 \otimes A_2 \otimes \dots \otimes A_r)^{-1} = A_1^{-1} \otimes A_2^{-1} \otimes \dots \otimes A_r^{-1}$;
- $\det(A_1 \otimes A_2 \otimes \dots \otimes A_r) = \det(A_1)^{c_1} \det(A_2)^{c_2} \dots \det(A_r)^{c_r}$, where $A_i \in \mathbb{R}^{k_i\times k_i}$, $c_i = \prod_{j\neq i} k_j$.

Inducing points on a regular grid

Let's put inducing inputs Z on a regular grid:

$$Z = Z^1 \times Z^2 \times \cdots \times Z^d,$$

where $Z^i \in \mathbb{R}^{m_i}$, $m = \prod_{i=1}^d m_i$.



Inducing points on a regular grid

Suppose that covariance function can be split over dimensions:

$$k(\boldsymbol{x}, \boldsymbol{y}) = k^1(x_1, y_1)k^2(x_2, y_2)\dots k^d(x_d, y_d).$$

E.g. for squared exponential:

$$k(\boldsymbol{x}, \boldsymbol{y}) = A\exp(-B\|\boldsymbol{x} - \boldsymbol{y}\|^2) =$$
$$\underbrace{A^{1/d}\exp(-B(x_1 - y_1)^2)}_{k^1(x_1, y_1)}\underbrace{A^{1/d}\exp(-B(x_2 - y_2)^2)}_{k^2(x_2, y_2)}\dots\underbrace{A^{1/d}\exp(-B(x_d - y_d)^2)}_{k^d(x_d, y_d)}$$

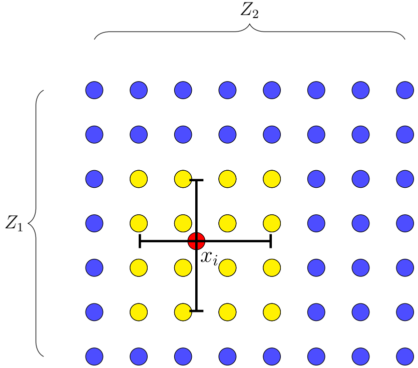
Then covariance matrix is given as Kronecker product:

$$K_{mm} = K^1_{m_1m_1} \otimes K^2_{m_2m_2} \otimes \dots \otimes K^d_{m_dm_d},$$
$$K^i_{m_im_i} = K^i(Z^i, Z^i) \in \mathbb{R}^{m_i \times m_i}.$$

Cubic convolution interpolation [Keys, 1981]

We need to estimate covariance between training and inducing inputs K_{mn}. In case of cubic convolution interpolation we have:

$$K_{mn} \approx K_{mm}W, \ \boldsymbol{k}_i \approx K_{mm}\boldsymbol{w}_i,$$
$$\boldsymbol{w}_i = \boldsymbol{w}_i^1 \otimes \boldsymbol{w}_i^2 \otimes \cdots \otimes \boldsymbol{w}_i^d.$$



Optimization criterion:

$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \log p(y_i|f_i) - \mathrm{KL}(q(\boldsymbol{u}) \| p(\boldsymbol{u}|Z, \boldsymbol{\theta})) \to \max_{\boldsymbol{\theta}, \boldsymbol{\mu}, \Sigma}.$$

If $\boldsymbol{k}_i = K_{mm}\boldsymbol{w}_i$, then

$$q(f_i) = \mathcal{N}(f_i | \boldsymbol{k}_i^T K_{mm}^{-1} \boldsymbol{\mu}, k_{ii} + \boldsymbol{k}_i^T K_{mm}^{-1}(\Sigma - K_{mm}) K_{mm}^{-1} \boldsymbol{k}_i) =$$
$$= \mathcal{N}(f_i | \boldsymbol{w}_i^T \boldsymbol{\mu}, k_{ii} + \boldsymbol{w}_i^T (\Sigma - K_{mm}) \boldsymbol{w}_i).$$

Second term:

$$\mathrm{KL}(q(\boldsymbol{u} \| p(\boldsymbol{u}|Z, \boldsymbol{\theta})) =$$
$$= -\frac{m}{2} - \frac{1}{2} \log\det K_{mm}^{-1}\Sigma + \frac{1}{2} \mathrm{tr} K_{mm}^{-1}\Sigma + \frac{1}{2} \boldsymbol{\mu}^T K_{mm}^{-1} \boldsymbol{\mu}.$$

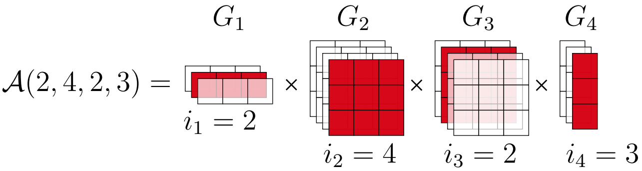
Tensor Train format [Oseledets, 2011]

Tensor Train format (TT format) gives a compact representation of multidimensional tensors. If $\mathcal{A} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_d}$, then

$$A(i_1, i_2, \ldots, i_d) = G[i_1]G[i_2]\ldots G[i_d],$$
$$G[i_k] \in \mathbb{R}^{r_k \times r_{k+1}},\ r_1 = r_{d+1} = 1.$$

Here $G[i_k]$ are TT cores and r_k – TT ranks.

TT format allows many linear algebra operations to perform efficiently.



A family for variational distribution $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}, \Sigma)$:

- $\boldsymbol{\mu}$ in TT format with fixed rank r;
- $\Sigma = \Sigma^1 \otimes \Sigma^2 \otimes \cdots \otimes \Sigma^d$.

Optimization criterion:

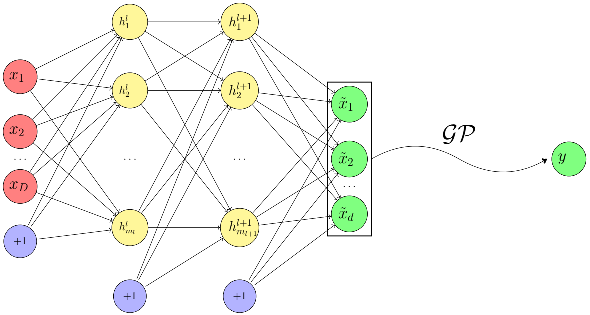
$$\log p(\boldsymbol{y}|X, Z, \boldsymbol{\theta}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(f_i)} \log p(y_i|f_i) - \frac{m}{2} + \frac{1}{2} \log \det K_{mm}^{-1} \Sigma - \\ - \frac{1}{2} \mathrm{tr} K_{mm}^{-1} \Sigma - \frac{1}{2} \boldsymbol{\mu}^T K_{mm}^{-1} \boldsymbol{\mu},$$

$$q(f_i) = \mathcal{N}(f_i|\boldsymbol{w}_i^T \boldsymbol{\mu}, k_{ii} + \boldsymbol{w}_i^T (\Sigma - K_{mm}) \boldsymbol{w}_i).$$

If m_0 is number of inducing inputs per dimension, then all calculations can be performed in $O(ndm_0 r^2 + dm_0 r^3 + dm_0^3)$.

TT-GP can be easily combined with deep net covariance function

$$k(\boldsymbol{x}, \boldsymbol{y}) = k(\mathrm{net}(\boldsymbol{x}; \boldsymbol{\eta}), \mathrm{net}(\boldsymbol{y}; \boldsymbol{\eta})).$$



Representation for Digits