Gaussian Processes and MC Dropout

What is a Gaussian Process

- Gaussian processes are stochastic processes - a collection of indexable random variables
- Given any finite collection of random variables from a GP, the joint distribution is Normal

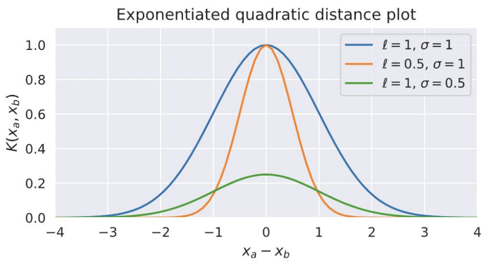
$$\{Y_x | x \in S\} \qquad \{Y_{x_1}, Y_{x_2}, \ldots Y_{x_n}\} \sim \text{MVN}$$

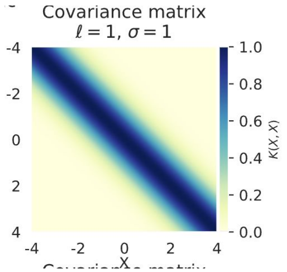
- Another way to view Gaussian Processes is a distribution over functions
- GPs can be used in a variety of applications as they are theoretically able to approximate any smooth function

What is a Gaussian Process

- To specify a GP, make a PSD symmetric matrix describing the covariance between $\mathbf{K} = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{bmatrix}$
    - Use a kernel function ***k*** mapping indices to covariances

- We can define kernel to be the exponential quadratic kernel $K(x_1, x_2) = \sigma^2 \exp\left(\frac{||x_1 - x_2||^2}{\lambda^2}\right)$





https://peterroelants.github.io/posts/gaussian-process-kernels/

GP Regression

- Suppose we want to model the relationship **y = f(x) + ε** where **ε ~ N(0,** σ^2**)** i.e.

  y ~ N(f(x), σ^2**)**

- We can use GP as a prior specification on **f(x)**

  f(x) ~ GP(0, K(x, x))

- After observing the x's, we get the MVN over the f(x)'s by instantiating the covariance matrix

  f(x) ~ N(0, K)

GP Regression

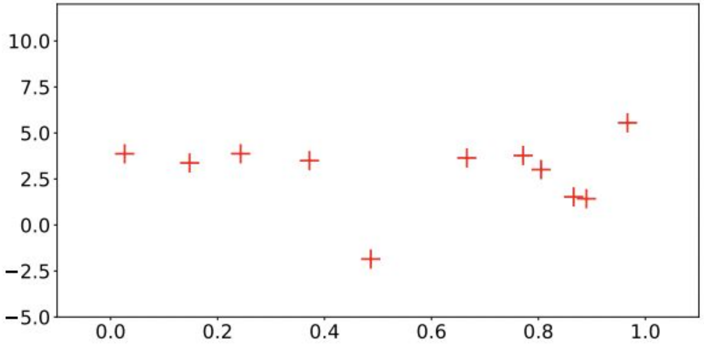
- How to make predictions on new indices $\mathbf{x}_*$? Get the joint distribution over labels and predictions $\mathbf{f}_*$

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_n^2 I & \mathbf{K}_* \\ \mathbf{K}_*^\top & \mathbf{K}_{**} \end{bmatrix}\right) \qquad \mathbf{K}_* = K(\mathbf{X}, \mathbf{X}_*) \text{ and } \mathbf{K}_{**} = K(\mathbf{X}_*, \mathbf{X}_*).$$

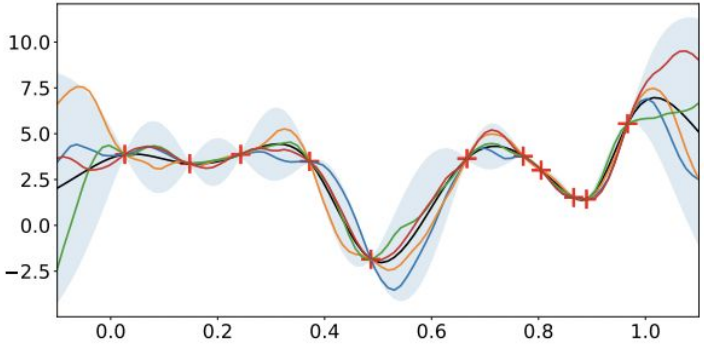
- Marginalize to get the predictive distribution

$$\mathbf{f}_* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}_* \sim \mathcal{N}\left(\bar{\mathbf{f}}_*, \operatorname{cov}(\mathbf{f}_*)\right)$$

$$\begin{aligned} \bar{\mathbf{f}}_* &\triangleq \mathbb{E}[\bar{\mathbf{f}}_* \mid \mathbf{X}, \mathbf{y}, \mathbf{X}_*] \\ &= \mathbf{K}_*^\top [\mathbf{K} + \sigma_n^2 I]^{-1} \mathbf{y} \ , \\ \operatorname{cov}(\mathbf{f}_*) &= \mathbf{K}_{**} - \mathbf{K}_*^\top [\mathbf{K} + \sigma_n^2 I]^{-1} \mathbf{K}_* \end{aligned}$$



(a) Data point observations



(b) Five possible functions by GPR

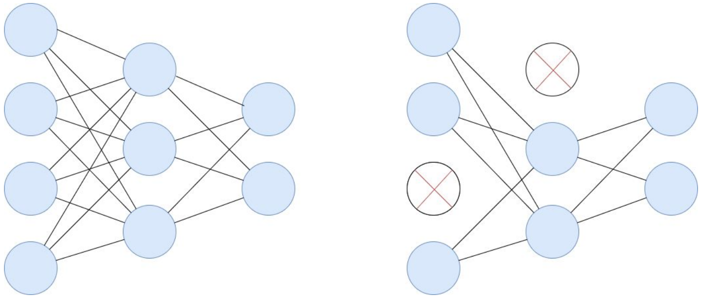
Remarks

- Gaussian Processes can be used in regression to fit any type of non-linearity
- It is a type of non-parametric model - number of parameters scales with the size of dataset
- Kernel hyperparameters can be tuned using MLE

MC Dropout

- In neural networks, dropout is used as regularization
- MC Dropout is just like regular NN dropout, but used also at test time
  - As a result, we can introduce uncertainty in the output
- Dropout in a general sense can be seen as adding random multiplicative noise to the input of each layer

$$\mathbf{B} = (\mathbf{A} \circ \xi)\theta, \quad \text{with } \xi_{i,j} \sim p(\xi_{i,j})$$



https://towardsdatascience.com/monte-carlo-dropout-7fd52f8b6571

Bayesian(?) interpretations of MC Dropout

Kingma, Salimans and Welling (2015) - Variational Dropout and the Local Reparameterization Trick

- Bayesian approximation using Variational Inference on a dropout network
- Uses the dropout rate as part of the variational parameters on the variational distribution
- An improper prior is chosen to prevent shrinkage effect on the weights

$$w_{jk}^{(h)} = \theta_{jk}^{(h)} \times \xi_{jk}^{(h)},\ \underbrace{\xi_{jk}^{(h)} \sim N(1, \alpha)}_{\text{Gaussian "dropout"}} \text{ with (improper log uniform) prior } p(\log |w_{jk}^{(h)}|) \propto c$$

$$q(w_{jk}^{(h)}) \equiv N\left(\theta_{jk}^{(h)}, \alpha(\theta_{jk}^{(h)})^2\right)$$

Bayesian(?) interpretations of MC Dropout

Gal and Ghahramani (2017) - Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

- Views the entire NN prediction object itself as a variational stochastic process, approximates another variational stochastic known as *Sparse Spectrum GP*
- Equivalency in objective functions between training a NN with dropout layers and SSGP

$$\underset{\text{Monte Carlo point masses}}{\overset{\text{NN prediction based on}}{\text{MC-Dropout NN}}} \xrightarrow[\text{approximates}]{\text{NN optimization}} \text{Sparse Spectrum GP} \xrightarrow[\text{"via VI"}]{\textit{approx.}} \text{GP}$$

$$\underbrace{w_{jk}^{(h)} \sim p^{(h)} N(m_{jk}^{(h)}, \sigma^2) + (1 - p^{(h)}) N(0, \sigma^2)}_{\text{a sparse spectrum GP takes this form and approximates a GP}} \overset{\sigma \to 0}{\approx} \underbrace{z_k^{(h)} m_{jk}^{(h)} + 0 \times (1 - z_{jk}^{(h)}),\ z_{jk}^{(h)} \sim bin(p^{(h)})}_{\text{a dropout NN where } z_{jk}^{(h)} \text{ is always Monte Carlo sampled}}$$

Bayesian(?) interpretations of MC Dropout

- Improper log uniform prior produces improper posteriors (HMG 2017)
- Issues with non-continuity in point mass approximation
- Mode collapse failure of the posterior distribution of dropout parameters (Osband 2016)

References

Schwartz 2021. https://github.com/pointOfive/Summer_2022_STA496H1/blob/main/files/GaussianProcesses.ipynb

Schwartz 2021. https://github.com/pointOfive/Summer_2022_STA496H1/blob/main/files/DropoutBayes.ipynb

Kingma, Salimans and Welling 2015. Variational Dropout and the Local Reparameterization Trick

Gal and Ghahramani 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Hron, Matthews and Ghahramani 2017. Variational Gaussian Dropout is not Bayesian

Wang 2020. An Intuitive Tutorial to Gaussian Processes Regression

Osband 2016. Risk vs. Uncertainty in Deep Learning: Bayes, Bootstrap, and the Dangers of Dropout