Gaussian Process for Regression

Fall 2019

Instructor: Shandian Zhe
zhe@cs.utah.edu
School of Computing

THE UNIVERSITY OF UTAH

Outline

- GP regression
- Training and prediction
- Connection to Bayesian neural networks

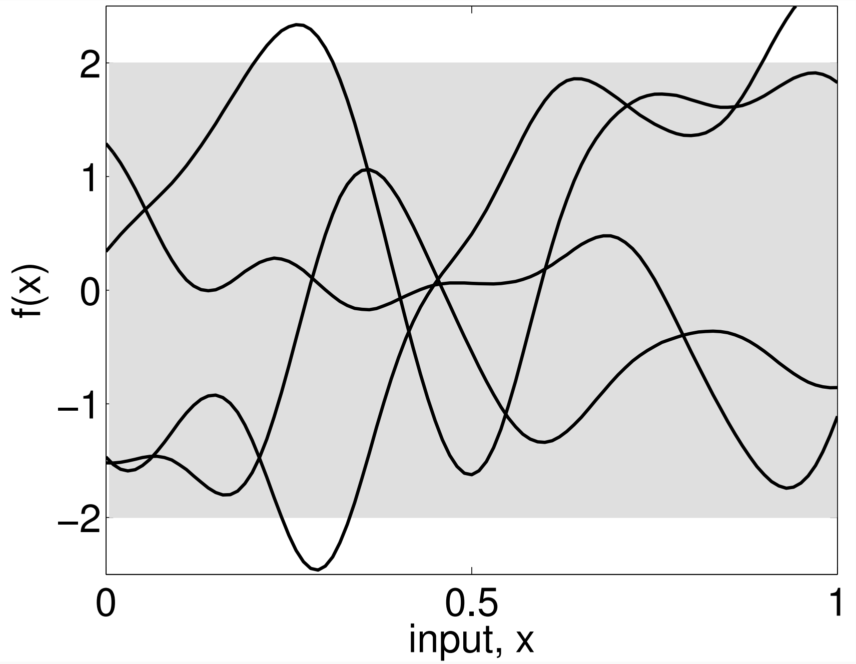
Outline

- GP regression
- Training and prediction
- Connection to Bayesian neural networks

Gaussian process priors

- Goal: how to assign a prior over functions?

$$f : \mathbb{R}^d \to \mathbb{R}$$



GPML book Fig.1.1

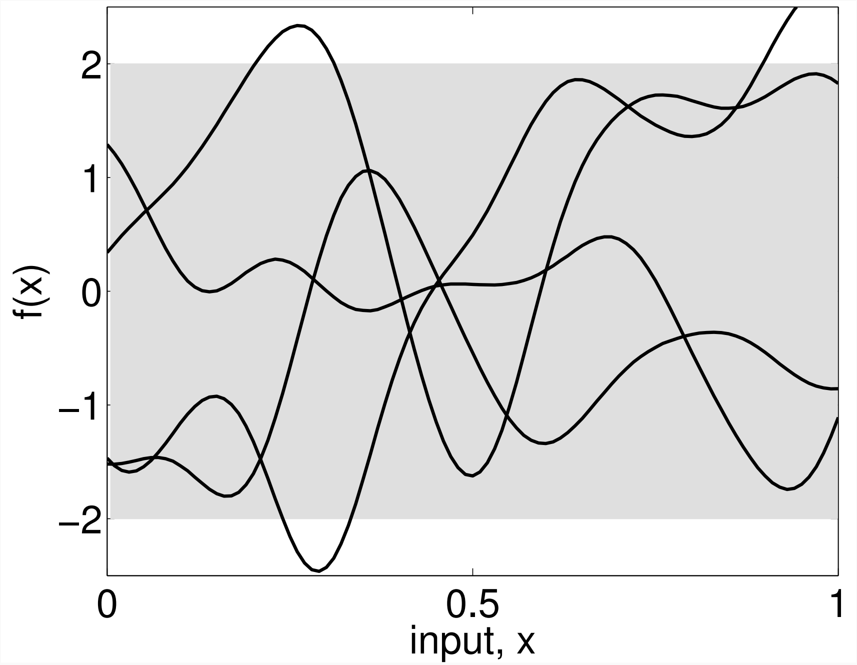
Gaussian process priors

We know how to place a prior over several random variables

$$p(\mathbf{z}) = p(z_1, \ldots, z_m)$$

But how to construct a prior to sample functions?

GP regression



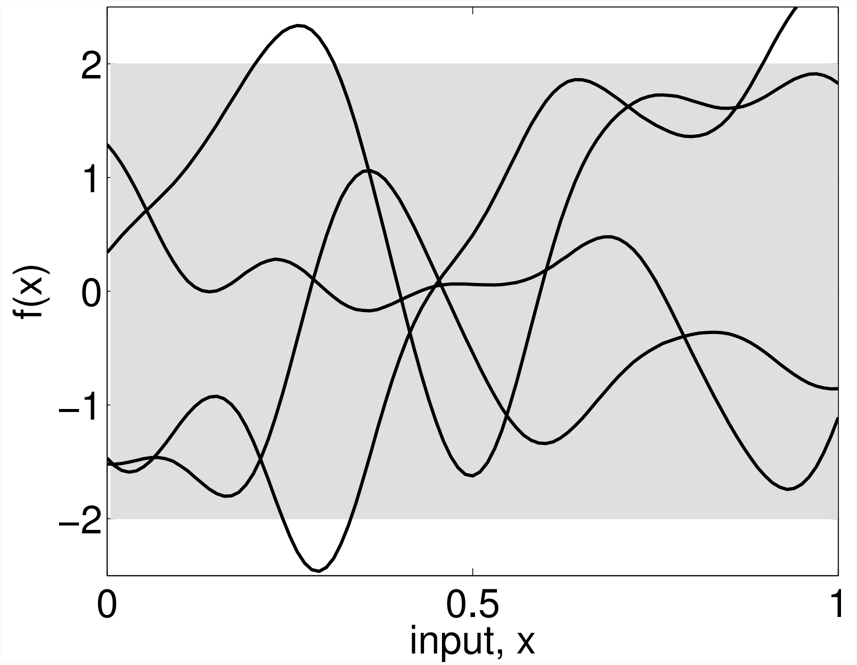
We can view function as a big table

Input	output
x_1	$f(x_1)$
x_2	$f(x_2)$
x_3	$f(x_3)$
…	…

We view each output as a random variable

We want to place a prior over all the function outputs!

GP regression



We can view function as a big table

Input	output
x_1	$f(x_1)$
x_2	$f(x_2)$
x_3	$f(x_3)$
…	…

Note that the possible inputs of a function are usually ***infinite*** and ***uncountable***, so rigorously speaking, we should not use integers to index the input

GP regression

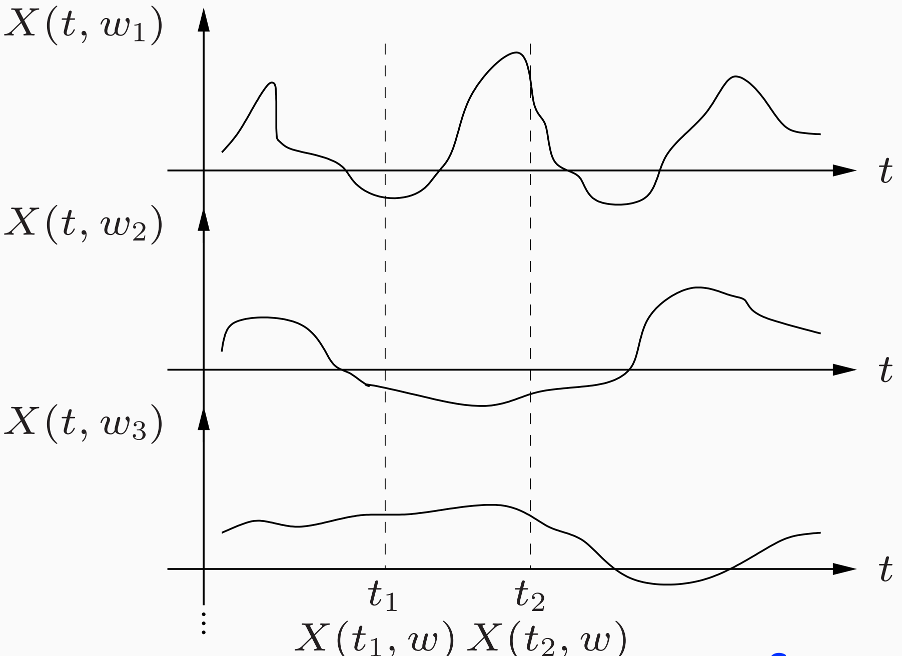
- That means, we need to assign the prior over the collection of all the function outputs (infinite, uncountable)
- Is it doable? Yes
- Such a prior is called a random process

$$\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^d\}$$

Two Ways to View a Random Process

- A random process can be viewed as a function $X(t, \omega)$ of two variables, time $t \in \mathcal{T}$ and the outcome of the underlying random experiment $\omega \in \Omega$
  - For fixed t, $X(t, \omega)$ is a random variable over Ω
  - For fixed ω, $X(t, \omega)$ is a deterministic function of t, called a *sample function*

Can be generalized to any continuous input



Source: Stanford Statistics Slides

GP prior

- What process do we use to sample function outputs?

$$\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^d\}$$

We use Gaussian process

A random process such that every finite collection of these random variables follow a multivariate Gaussian distribution.

GP prior

Given any finite set of inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$

The corresponding function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$

follows a multivariate Gaussian distribution

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f} | \mathbf{0}, \mathbf{\Sigma})$$

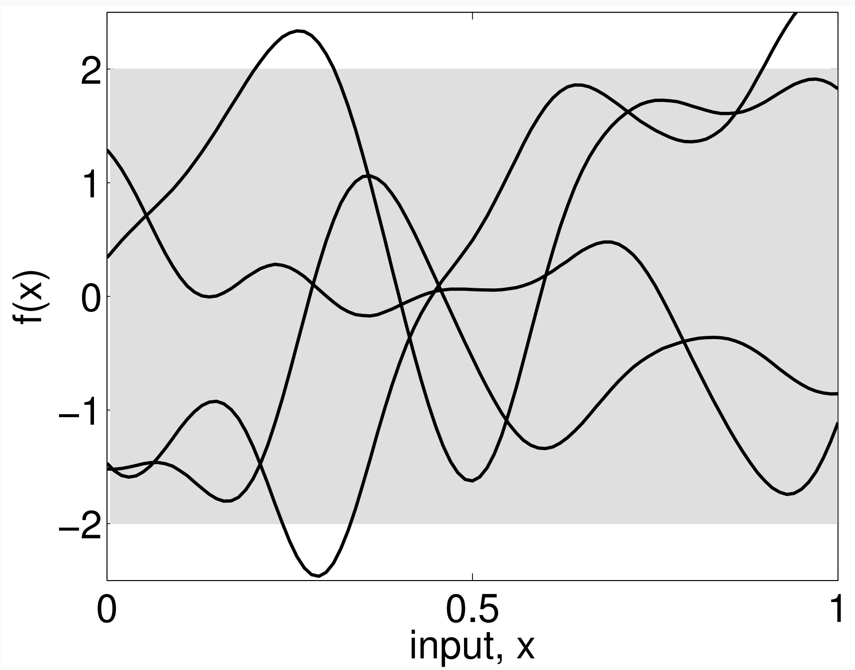
$$\mathbf{\Sigma} = k(\mathbf{X}, \mathbf{X})$$ Kernel matrix of the inputs

$$\begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \ldots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \ldots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \ldots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

GP prior

- Kernel function measures the similarity of two inputs

e.g., RBF $$k(\mathbf{x}_j, \mathbf{x}_j) = \exp(-\frac{1}{\tau}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

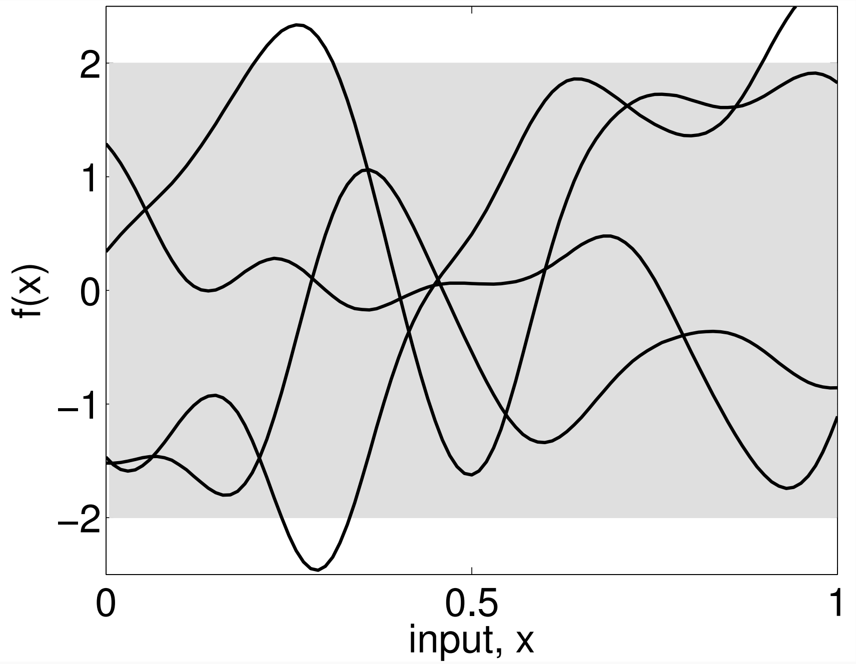


It essentially implies that the closer the inputs, the more correlated the function outputs. It describes the function smoothness in the probabilistic context

GP prior

- Kernel function measures the similarity of two inputs

e.g., RBF $$k(\mathbf{x}_j, \mathbf{x}_j) = \exp(-\frac{1}{\tau}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$



There are numerous ways to define your kernel function. Different kernel functions defines different ways to measure the similarity!

GP regression

- In practice, we will never need to sample the whole function, because the training data are always finite.
- Given the training data,

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \quad \mathbf{y} = [y_1, \ldots, y_N]^\top$$

How to construct our probabilistic model to sample the data?

GP regression

- Given the training data,

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \quad \mathbf{y} = [y_1, \ldots, y_N]^\top$$

How to construct our probabilistic model to sample the data?

- We first sample the function values at the inputs

$\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$ from the multivariate Gaussian prior (this is a finite projection of the GP prior)

$$\mathbf{f} \sim \mathcal{N}\big(\mathbf{f} | \mathbf{0}, k(\mathbf{X}, \mathbf{X})\big)$$

GP regression

- Given the training data,

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \qquad \mathbf{y} = [y_1, \ldots, y_N]^\top$$

How to construct our probabilistic model to sample the data?

- Given the function values ***f***, we sample the observed outputs.

$$\mathbf{y}|\mathbf{f} \sim p(\mathbf{y}|\mathbf{f})$$

For regression task (continuous output), we usually use Gaussian likelihood,

$$p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I})$$

Observation are corrupted by some Gaussian white noise

GP regression

- The joint probability

$\boldsymbol{\theta}$: noise variance and kernel parameters

$$p(\mathbf{y}, \mathbf{f} | \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y} | \mathbf{f}, \sigma^2 \mathbf{I}) \mathcal{N}\big(\mathbf{f} | \mathbf{0}, k(\mathbf{X}, \mathbf{X})\big)$$

- We can marginalize out latent function values

$$p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\big(\mathbf{y} | \mathbf{0}, k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\big)$$

GP regression: kernel

- Requirement on kernel function: for any finite number of inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$

$$\begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \ldots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \ldots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \ldots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

must be semi-positive definite!

Mercer's condition
(discrete version)

GP regression: kernel examples

- Linear kernel: $k(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T\mathbf{z}$
- Polynomial kernel of degree *d*: $k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z})^d$
- Polynomial kernel up to degree *d*: $k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + c)^d$ (c>0)
- RBF $K_{rbf}(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{z}||^2}{c}\right)$
- Periodic kernel $\sigma_f^2 \exp\left(-\frac{2}{\ell^2}\sin^2\left(\pi\frac{x-x'}{p}\right)\right)$
- Matern kernel
- …

GP regression: kernel examples

- Each kernel function corresponds to a (possibly) high-dimensional, nonlinear feature mapping

$$\psi : \mathbb{R}^k \to \mathbb{R}^d$$

often times: $\begin{cases} d \gg k \\ d = \infty \end{cases}$

$$k(\mathbf{x}_1, \mathbf{x}_2) = \psi(\mathbf{x}_1)^\top \psi(\mathbf{x}_2)$$

Kernel function a cheap way to compute inner-product of high-dimensional feature vectors!

GP regression: linear model view

- Given the training data,

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \qquad \mathbf{y} = [y_1, \ldots, y_N]^\top$$

We first sample an (infinite dimensional) weight vector

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{I})$$

$$y_n \sim \mathcal{N}(y_n|\mathbf{w}^\top \psi(\mathbf{x}_n), \sigma^2 \mathbf{I})$$

Marginalize out ***w***

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\big(\mathbf{y}|\mathbf{0}, k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\big)$$ Why?

Outline

- GP Regression
- Training and prediction
- Connection to Bayesian neural networks

Training and prediction

$\boldsymbol{\theta}$: noise variance and kernel parameters

$$p(\mathbf{y}, \mathbf{f} | \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y} | \mathbf{f}, \sigma^2 \mathbf{I}) \mathcal{N}\big(\mathbf{f} | \mathbf{0}, k(\mathbf{X}, \mathbf{X})\big)$$

We can perform EM algorithm to jointly estimate the posterior of ***f*** and hyper-parameters

However, in practice, we often do type II MLE

$$p(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}\big(\mathbf{y} | \mathbf{0}, k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\big)$$

$$\max_{\boldsymbol{\theta}} \ \log \mathcal{N}\big(\mathbf{y} | \mathbf{0}, k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\big)$$

Training and prediction

- How to make a prediction? conditional Gaussian!

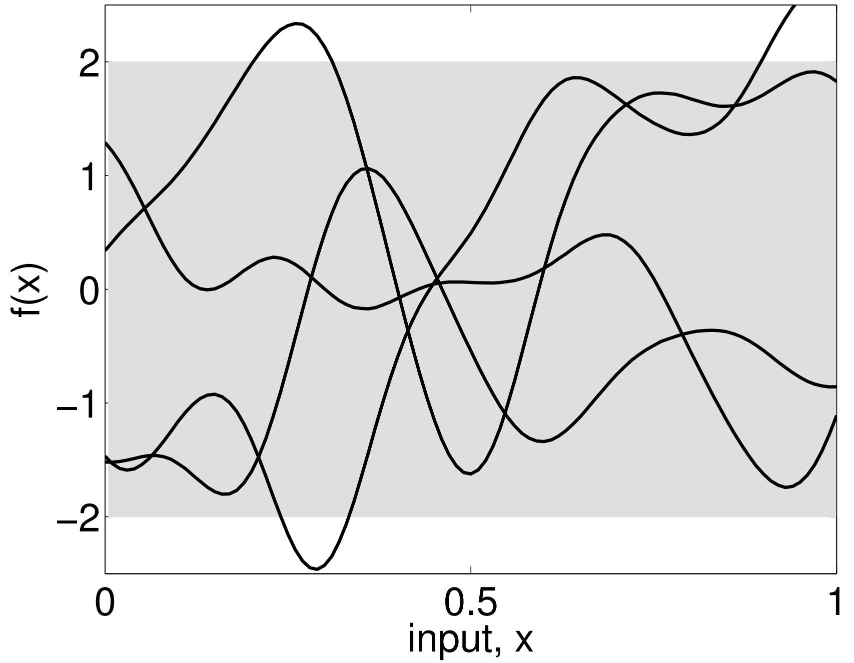
$$\begin{bmatrix} f(\mathbf{x}^*) \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} f(\mathbf{x}^*) \\ \mathbf{y} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}^*, \mathbf{x}^*) & k(\mathbf{x}^*, \mathbf{X}) \\ k(\mathbf{X}, \mathbf{x}*) & k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} \end{bmatrix} \right)$$

We can easily compute $p\big(f(\mathbf{x}^*)|\mathbf{y}\big)$

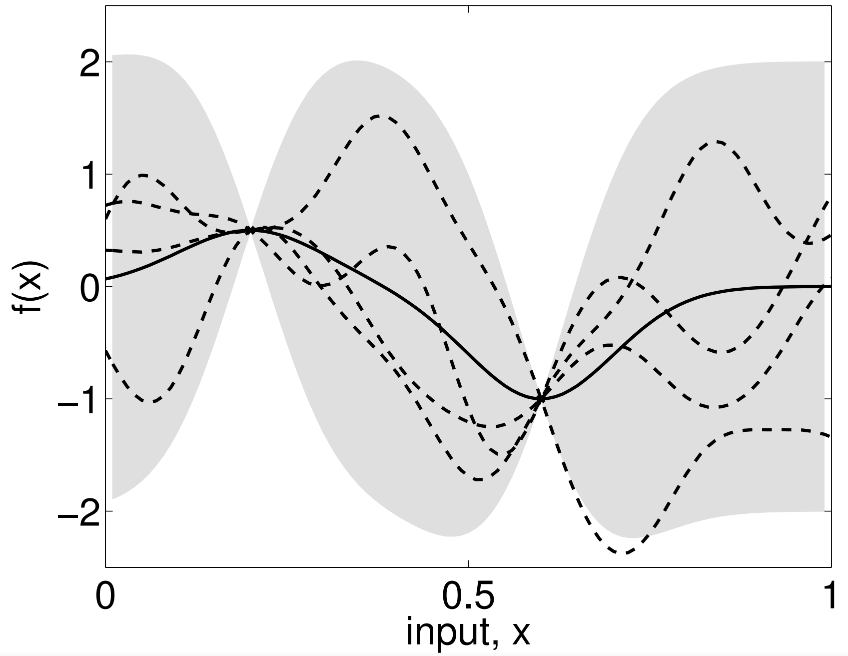
$$\mathcal{N}\big(f(\mathbf{x}^*)|\underbrace{k(\mathbf{x}^*, \mathbf{X})(k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1}\mathbf{y}}_{\text{Predictive mean}}, \underbrace{k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{X})(k(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I})^{-1} k(\mathbf{X}, \mathbf{x}^*)}_{\text{Predictive variance}}\big)$$

Training and prediction

Prior of the functions



Posterior of the functions

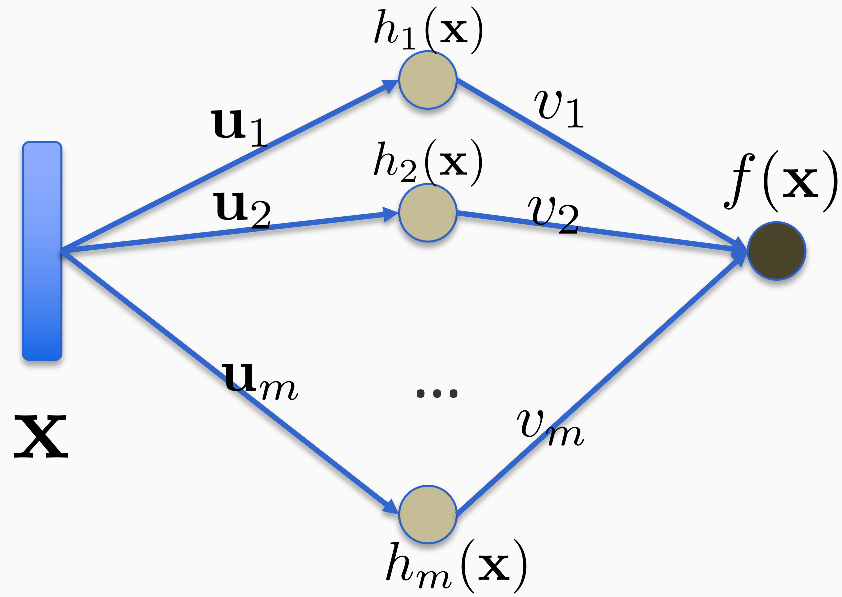


Outline

- GP Regression
- Training and prediction
- Connection to Bayesian neural networks

Connection to BNNs

- A famous conclusion discovered by Radford M. Neal (1994)
- Consider an NN with only one hidden layer

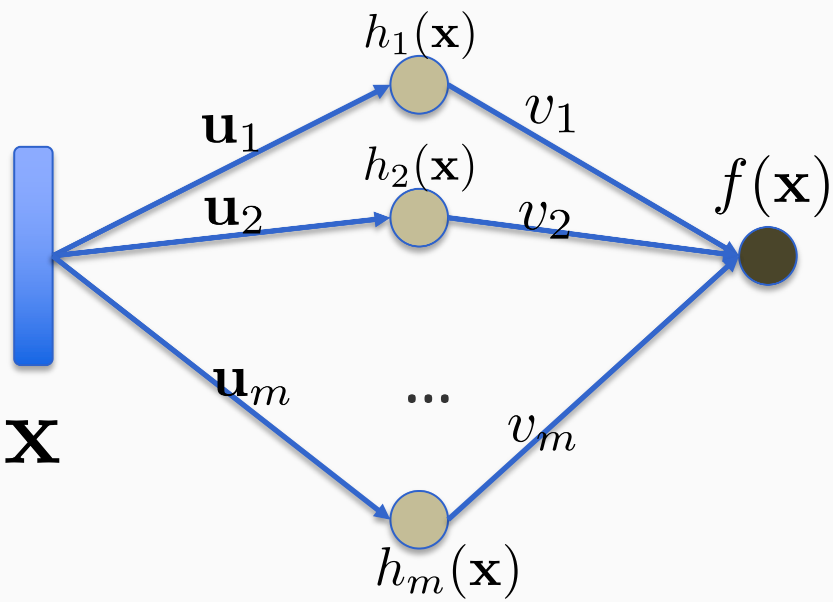


$$h_j(\mathbf{x}) = \sigma(\mathbf{u}_j^\top \mathbf{x}) \ (1 \leq j \leq m)$$

$$f(\mathbf{x}) = \sum_{j=1}^{m} v_j h_j(\mathbf{x})$$

Connection to BNNs

- A famous conclusion discovered by Radford M. Neal (1994)
- Consider an NN with only one hidden layer

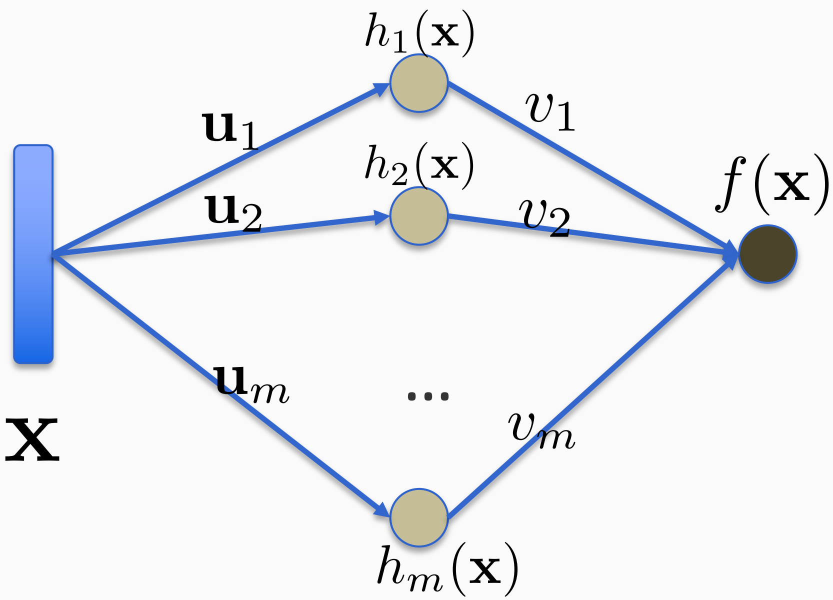


activation function: tanh, sigmoid, ...

$$h_j(\mathbf{x}) = \sigma(\mathbf{u}_j^\top \mathbf{x}) \quad (1 \le j \le m)$$

$$f(\mathbf{x}) = \sum_{j=1}^{m} v_j h_j(\mathbf{x})$$

Connection to BNNs



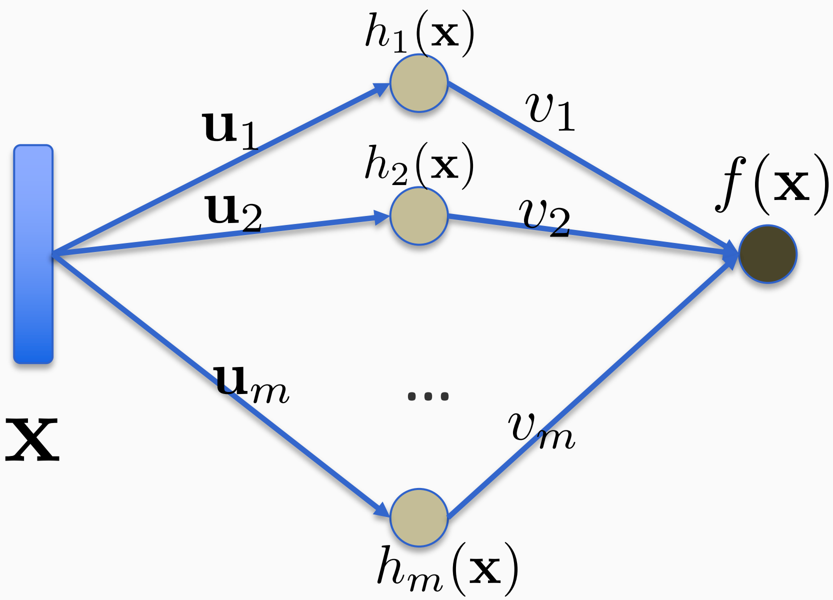
$$h_j(\mathbf{x}) = \sigma(\mathbf{u}_j^\top \mathbf{x}) \ (1 \leq j \leq m)$$

$$f(\mathbf{x}) = \sum_{j=1}^{m} v_j h_j(\mathbf{x})$$

We assign the same prior each $\mathbf{u}_j$ bounded variance

We assign the same prior each v_j with 0 mean and a variance $\frac{\omega^2}{m}$

Connection to BNNs



activation function: tanh, sigmoid, ….

$$h_j(\mathbf{x}) = \sigma(\mathbf{u}_j^\top \mathbf{x}) \ (1 \leq j \leq m)$$

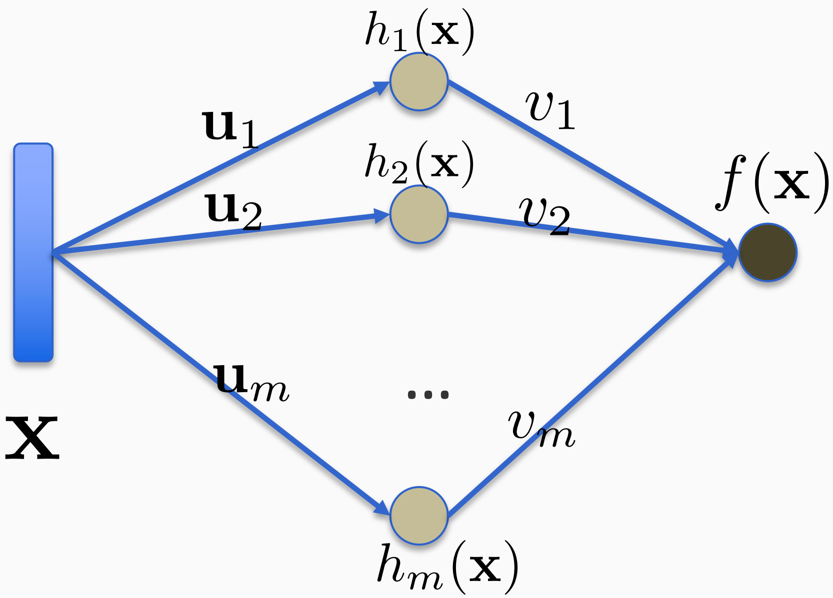
$$f(\mathbf{x}) = \sum_{j=1}^{m} v_j h_j(\mathbf{x})$$

We assign the same prior each $\boldsymbol{u}_j$ bounded variance

We assign the same prior each v_j with 0 mean and a variance $\frac{\omega^2}{m}$

Then we can prove that when $m \to \infty$, $f(\mathbf{x})$ follows a GP prior

Proof sketch



activation function: tanh, sigmoid, ...

$$h_j(\mathbf{x}) = \sigma(\mathbf{u}_j^\top \mathbf{x}) \ (1 \le j \le m)$$

$$f(\mathbf{x}) = \sum_{j=1}^{m} v_j h_j(\mathbf{x})$$

We assign the same prior each $\boldsymbol{u}_j$ bounded variance

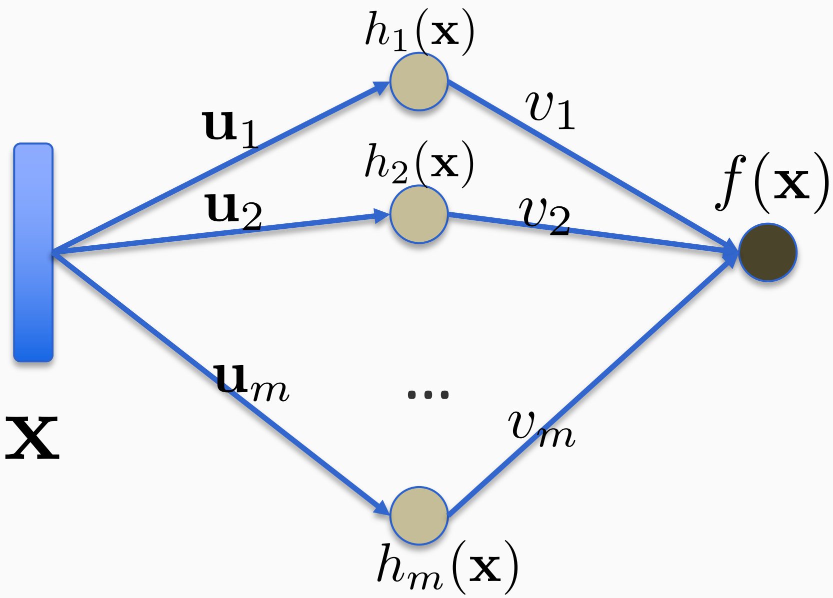
We assign the same prior each v_j with 0 mean and a variance $\frac{\omega^2}{m}$

$\{\sqrt{m} v_1 h_1(\mathbf{x}), \ldots, \sqrt{m} v_m h_m(\mathbf{x})\}$ are IID with 0 mean and constant variance (to m)

$$f(\mathbf{x}) = \sqrt{m} \cdot \frac{1}{m} \sum_{j=1}^{m} \sqrt{m} v_j h_j(\mathbf{x})$$ Scaled average of IID variables

From Central Limit theorem, $f(\mathbf{x})$ follows a Gaussian distribution when $m \to \infty$

Proof sketch



activation function: tanh, sigmoid, ...

$$h_j(\mathbf{x}) = \sigma(\mathbf{u}_j^\top \mathbf{x}) \ (1 \le j \le m)$$

$$f(\mathbf{x}) = \sum_{j=1}^{m} v_j h_j(\mathbf{x})$$

We assign the same prior each $\boldsymbol{u}_j$ bounded variance

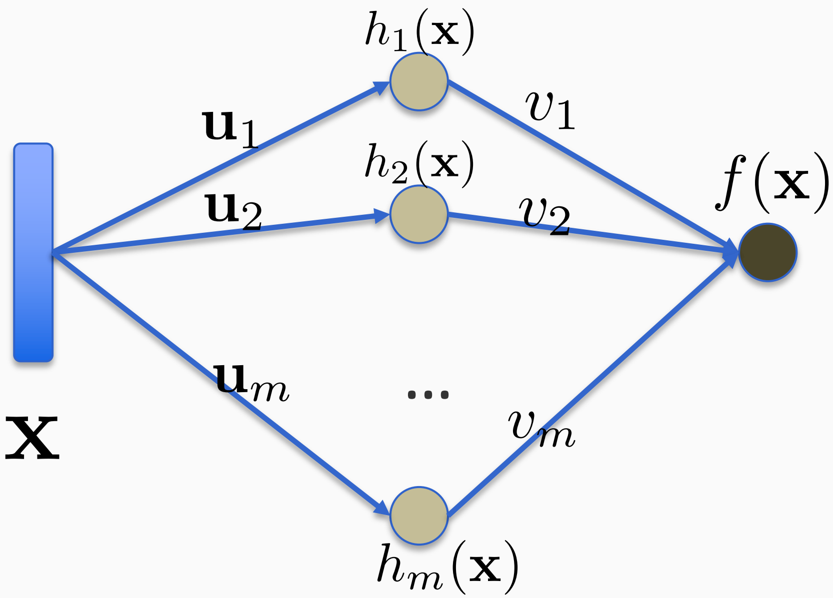
We assign the same prior each v_j with 0 mean and a variance $\frac{\omega^2}{m}$

From Central Limit theorem, $f(\mathbf{x})$ follows a Gaussian distribution when $m \to \infty$

The result can be generated for an arbitrary set of inputs $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

$[f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$ follows a multivariate Gaussian distribution when $m \to \infty$

Proof sketch



activation function: tanh, sigmoid, ….

$$h_j(\mathbf{x}) = \sigma(\mathbf{u}_j^\top \mathbf{x}) \; (1 \leq j \leq m)$$

$$f(\mathbf{x}) = \sum_{j=1}^{m} v_j h_j(\mathbf{x})$$

We assign the same prior each $\boldsymbol{u}_j$ bounded variance

We assign the same prior each v_j with 0 mean and a variance $\frac{\omega^2}{m}$

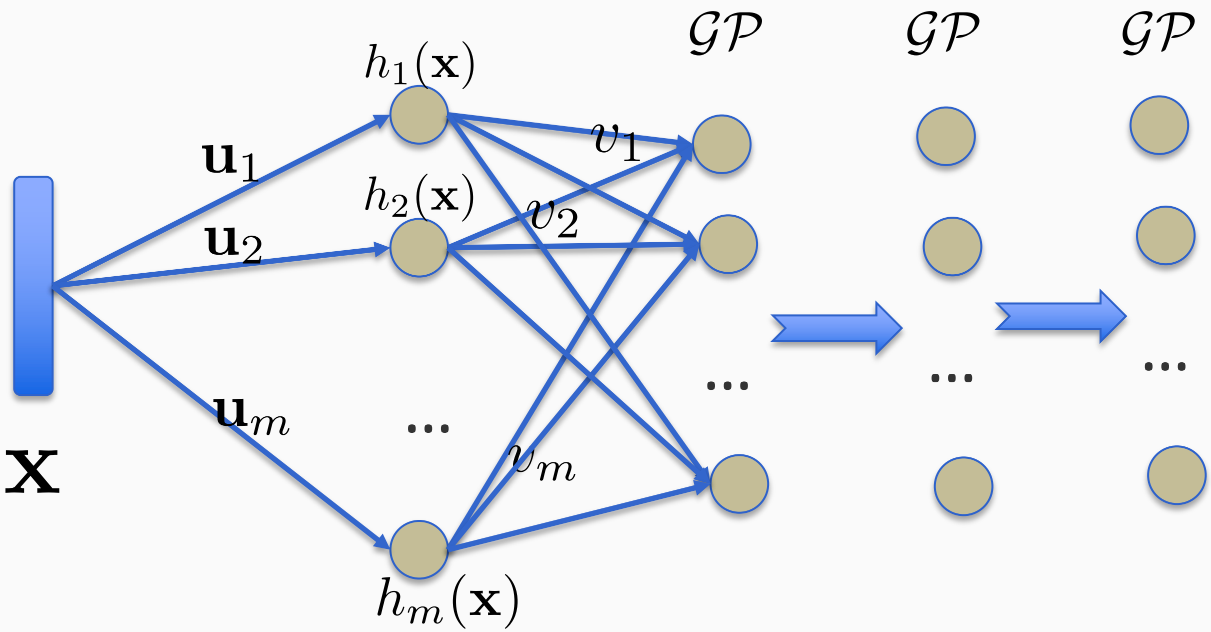
From Central Limit theorem, $f(\mathbf{x})$ follows a Gaussian distribution when $m \to \infty$

The result can be generated for an arbitrary set of inputs $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

That means $f(\cdot)$ follows a GP prior

Connection to BNNs

- Can be extended to deep NNs (Lee et. al. 2017)



Lee, Jaehoon, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. "**Deep neural networks as Gaussian Processes.**" *arXiv preprint arXiv:1711.00165* (2017).

Summary

- GP regression is a very powerful nonparametric model for function estimation
- Does not assume function forms
- Two views of GP priors
- Close-form predictive distribution
- Profound connections to BNNs