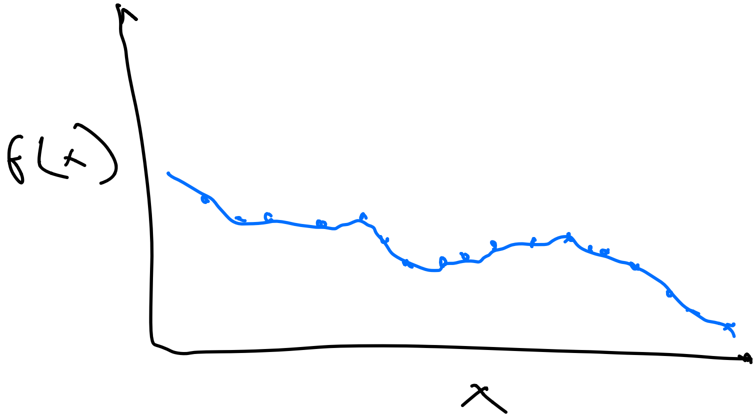
Gaussian processes

Idea: specify a prior $p(f)$

$$y_i = f(x_i) + \varepsilon_i$$
$$\varepsilon_i \sim N(0, \sigma^2)$$

$$f = [f_1, f_2, f_3, \ldots]$$



GP:

any finite subset of points $\{f_i\}$ are jointly Gaussian

Priors on functions

$$f_i = \omega^T \phi(x_i)$$

$$\omega \sim \mathcal{N}(0, \Sigma)$$

$x_i \in \mathbb{R}^D$

$f_i \in \mathbb{R}$

$\phi : \mathbb{R}^D \to \mathbb{R}^H$

$\omega \in \mathbb{R}^H$

for inputs $x_1, \ldots, x_N$ as matrices

$$\underset{N\times 1}{f} = \underset{N\times H}{\Phi}\, \underset{H\times 1}{\omega}$$

Q: what is $p(f)$?

$$E[f] = \Phi E[\omega] = 0$$

$$\mathrm{Cov}(f) = E[ff^T] - \cancel{E[f]}^{\,0}E[f]^T$$

$$= \Phi E[\omega\omega^T]\Phi^T$$

$$= \Phi \Sigma \Phi^T \equiv K$$

Kernels

What is this $K = \Phi \Sigma \Phi^T = \mathrm{Cov}(f)$?

$$K_{ij} = \underbrace{\mathrm{Cov}(f(x_i), f(x_j))}_{\text{output cov.}} = \underbrace{\phi(x_i)^T \Sigma \phi(x_j)}_{k(x_i, x_j)}$$

$$f \sim \mathcal{N}(0, K)$$

Kernel „trick"

Any $k(x, x')$ can be used as long as K is P.S.D.
↳ $\forall c,\ cKc \geq 0$

e.g. $\boxed{k(x, x') = e^{\frac{-\|x - x'\|_2^2}{2\ell^2}}}$ } squared exponential kernel

Conditioning on data

Suppose we observe some x, f examples, and we would like to predict at x^*:

$$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \overset{C}{K(x,x)} & \overset{A^T}{K(x,x^*)} \\ \overset{A}{K(x^*,x)} & \overset{D}{K(x^*,x^*)} \end{bmatrix} \right)$$

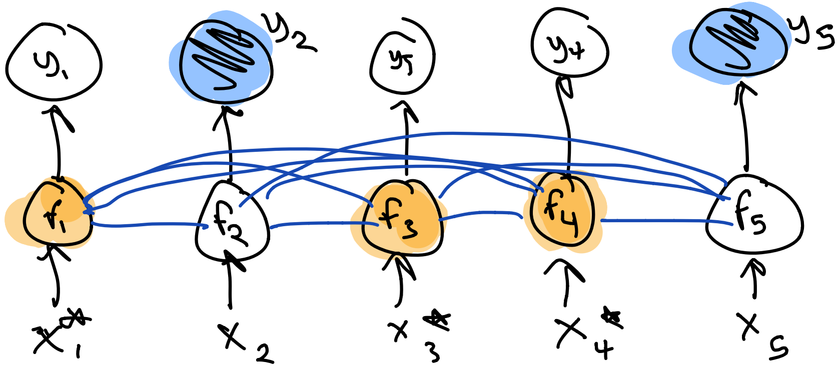
$$f^* \mid x^*, \underset{\substack{\uparrow \\ \text{or } y}}{f}, x \sim \mathcal{N}\left(AC^{-1}f,\ D - AC^{-1}A^T\right)$$

Gaussian properties

Noisy data

Include a likelihood $p(y_i \mid f_i)$, e.g. $y_i \sim N(f_i, \sigma^2)$

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \underbrace{K(x,x) + \sigma^2 I}_{C} & K(x, x^*) \\ K(x^*, x) & K(x^*, x^*) \end{bmatrix} \right)$$



In feedforward networks (Neal, 1996)

$$f(x_i) = a + W^T g(b + V^T x_i)$$

$\in \mathbb{R}$; scalar ; $\omega \in H \times 1$; nonlinearity ; $\mathbb{R}^D$

$$= a + \sum_{h=1}^{H} \omega_h \left[g(b + v_h^T x_i) \right]$$

scalar ; $\in \mathbb{R}^D$

$a \sim N(0, \sigma_a^2)$

$\omega_h \sim N(0, \sigma_\omega^2)$

$b \sim p(b)$

$v_h \overset{iid}{\sim} p(v)$

Compute moments:

$$E[f(x)] = E[a] + \sum_h E[\omega_h] E[g(\dots)] = 0$$

with $E[a] = 0$ and $E[\omega_h] = 0$.

$$\underbrace{E[f(x) f(x')]}_{= Cov(x, x')} = \sigma_a^2 + \sum_h \sigma_\omega^2 E\left[g(b + v_h^T x) g(b + v_h^T x') \right]$$

$$= \sigma_a^2 + H \sigma_\omega^2 E\left[g(b + v^T x) g(b + v^T x') \right], \; v \overset{iid}{\sim} p(v)$$

Now let $p(w) = \mathcal{N}\left(w \mid 0, \frac{s^2}{H}\right)$.

$$\mathrm{Cov}(x, x') = \sigma_a^2 + s^2 \overbrace{\mathbb{E}_{p(b)p(v)}\left[g(b+v^T x)\, g(b+v^T x')\right]}^{\text{sometimes analytic} \;\to\; \text{„arc-cosine kernel"}}$$

prior

Let $g(\cdot)$ be bounded (e.g. sigmoid, tanh)

CLT: a $H \to \infty$, this becomes Gaussian

(sample avg. i.i.d. r.v.)

For small H: probably not a GP

large H: probably close

More layers?

Recursion:

$$\begin{bmatrix} z^{l-1} \sim GP \\ h^{l-1} = g\left(z^{l-1}\right) \\ z^{l} = b^{l} + \sum_j \underbrace{w_j^{l}}_{w_j^{l} \sim N\left(0, \frac{\sigma_w^2}{H^l} I\right)} h_j^{l-1} \end{bmatrix}$$

$$\Rightarrow E\left[z^{l}(x)\, z^{l}(x')^{T}\right] = \sigma_b^2 + \sigma_w^2 E_{p(z^{l-1})}\left[g\left(z^{l-1}(x)\right) g\left(z^{l-1}(x')\right)\right]$$

Needs work to show it is <u>actually</u> a GP!

Simplified result:

Let $f_\theta(x)$ be a network prior $p(\theta)$

Define:

$$m(x) = E_{p(\theta)}\left[f_\theta(x)\right]$$ (often zero)

$$k(x, x') = E_{p(\theta)}\left[\left(f_\theta(x) - m(x)\right)\left(f_\theta(x') - m(x')\right)\right]$$

Now, use the $GP\left(m(\cdot), k(\cdot, \cdot)\right)$!

Is this good?

- Deep learning „works" because of hierarchies of features.
- GPs w/ „complicated" covariance functions don't perform as well
- May be contrary as much as useful!