Some More Kernels + Deep Learning

CPSC 532S: Modern Statistical Learning Theory

14 March 2022

cs.ubc.ca/~dsuth/532S/22/

Admin

- A3 is up; due **next Friday the 25th**
- This week only, Thursday office hours are instead **Wednesday 10-11am**
  - both in X563 and on the class Zoom
- Project proposals are due by Wednesday night
  - An informal paragraph on Piazza: just tell me what you want to do
  - Again, the scope of these is meant to be *small*
    - A lit survey doesn't require fully understanding the proofs or anything
    - An "extension" could be just reading the proofs and talking about when the assumptions hold, etc

Last time

$k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

$\mathcal{H}_k \ni f: \mathcal{X} \to \mathbb{R}$

$k(x, \cdot) \in \mathcal{H}_k$

$\langle f, k(x, \cdot) \rangle_{\mathcal{H}_k} = f(x)$

- We defined the RKHS for a given kernel k
- Representer theorem:

increasing function

$$\text{argmin}_{f \in \mathcal{H}} L(f(x_1), \ldots, f(x_n)) + R(\|f\|_{\mathcal{H}}) \in \text{span}(\{k(x_i, \cdot)\}_{i=1}^n)$$

- We can kernelize any algorithm that only depends on $x_i \cdot x_j$
- Applied previous bounds on generalization gap / suboptimality to kernels
  - Dependence on $\|x\|$ becomes $\sqrt{k(x, x)}$
  - Dependence on $\|w\|$ becomes $\|f\|_{\mathcal{H}}$

Universal kernels

- What about that $L_S(f)$ or $\inf_{\|f\|_{\mathcal{H}_k}} L_{\mathscr{D}}(f)$ term?

"Kernel": RKHS kernel (Mercer kernel) K symmetric, psd function

kernel density estimation: $K(x,x') \in \mathbb{R}$ symmetry ✓ psd ✗

need $\int K(x,x')\,dx = 1$

often $K(x,x') \geq 0$

Ker(A) linear map A - null space

Convolutional kernels $\neq$ Bayesians: kernel of a pdf

$\neq$ CUDA kernel $\neq$ Linux kernel

$\neq$ popcorn kernel

Universal kernels

- What about that $L_S(f)$ or $\inf\limits_{\|f\|_{\mathscr{H}_k}} L_{\mathscr{D}}(f)$ term?

continuous functions $\mathcal{X} \to \mathbb{R}$

- A continuous kernel on a compact metric space $\mathscr{X}$ is **universal** if $\mathscr{H}_k$ is dense in $C(\mathscr{X})$:
  for every continuous $g : \mathscr{X} \to \mathbb{R}$, every $\varepsilon > 0$, there is an $f \in \mathscr{H}_k$
  with $\|f - g\|_\infty = \sup\limits_{x \in \mathscr{X}} |f(x) - g(x)| \leq \varepsilon$

e.g. $\mathcal{X} = \{ x \in \mathbb{R}^d : \|x\| \leq R \}$

Universal kernels

- What about that $L_S(f)$ or $\inf_{\|f\|_{\mathcal{H}_k}} L_{\mathcal{D}}(f)$ term?
- A continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}_k$ is dense in $C(\mathcal{X})$: for every continuous $g : \mathcal{X} \to \mathbb{R}$, every $\varepsilon > 0$, there is an $f \in \mathcal{H}_k$ with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \le \varepsilon$
  - If $\mathcal{X}$ is a topological space *not* generated by a metric, there is no universal kernel (Steinwart/Christmann exercise 4.13)

Universal kernels

- What about that $L_S(f)$ or $\inf_{\|f\|_{\mathcal{H}_k}} L_{\mathcal{D}}(f)$ term?
- A continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}_k$ is dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, every $\varepsilon > 0$, there is an $f \in \mathcal{H}_k$
  with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \leq \varepsilon$
  - If $\mathcal{X}$ is a topological space *not* generated by a metric, there is no universal kernel (Steinwart/Christmann exercise 4.13)
  - Separates compact sets: if $X_1 \cap X_2 = \emptyset$ are compact subsets of $\mathcal{X}$,
    there's an $f \in \mathcal{H}_k$ with $f(x) > 0$ for $x \in X_1$, $f(x) < 0$ for $x \in X_2$ (so VCdim = ∞)

Universal kernels

- What about that $L_S(f)$ or $\inf_{\|f\|_{\mathcal{H}_k}} L_{\mathcal{D}}(f)$ term?
- A continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}_k$ is dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, every $\varepsilon > 0$, there is an $f \in \mathcal{H}_k$
  with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \le \varepsilon$
  - If $\mathcal{X}$ is a topological space *not* generated by a metric, there is no universal kernel (Steinwart/Christmann exercise 4.13)
  - Separates compact sets: if $X_1 \cap X_2 = \emptyset$ are compact subsets of $\mathcal{X}$,
    there's an $f \in \mathcal{H}_k$ with $f(x) > 0$ for $x \in X_1$, $f(x) < 0$ for $x \in X_2$ (so VCdim = ∞)
    - Implies that as $B \to \infty$, get $\inf_{\mathcal{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathcal{H}_{k,B}} L_{\mathcal{D}}(f) \to$ Bayes error if $\mathcal{D}$ has compact support

$\mathcal{H}_{k,B} = \{ f \in \mathcal{H}_k : \|f\|_{\mathcal{H}_k} \le B \}$

Universal kernels

- What about that $L_S(f)$ or $\inf_{\|f\|_{\mathcal{H}_k}} L_{\mathcal{D}}(f)$ term?
- A continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}_k$ is dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, every $\varepsilon > 0$, there is an $f \in \mathcal{H}_k$
  with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \leq \varepsilon$
  - If $\mathcal{X}$ is a topological space *not* generated by a metric, there is no universal kernel (Steinwart/Christmann exercise 4.13)
  - Separates compact sets: if $X_1 \cap X_2 = \emptyset$ are compact subsets of $\mathcal{X}$,
    there's an $f \in \mathcal{H}_k$ with $f(x) > 0$ for $x \in X_1$, $f(x) < 0$ for $x \in X_2$ (so VCdim = ∞)
    - Implies that as $B \to \infty$, get $\inf_{\mathcal{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathcal{H}_{k,B}} L_{\mathcal{D}}(f) \to$ Bayes error if $\mathcal{D}$ has compact support
  - Can show universality via Stone-Weierstrass (more later), or Fourier properties

$K(x, y) = \underbrace{\psi}(x - y)$

Universal kernels

- What about that $L_S(f)$ or $\inf_{\|f\|_{\mathcal{H}_k}} L_{\mathcal{D}}(f)$ term?
- A continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}_k$ is dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, every $\varepsilon > 0$, there is an $f \in \mathcal{H}_k$
  with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \leq \varepsilon$
  - If $\mathcal{X}$ is a topological space *not* generated by a metric, there is no universal kernel (Steinwart/Christmann exercise 4.13)
  - Separates compact sets: if $X_1 \cap X_2 = \emptyset$ are compact subsets of $\mathcal{X}$,
    there's an $f \in \mathcal{H}_k$ with $f(x) > 0$ for $x \in X_1$, $f(x) < 0$ for $x \in X_2$ (so VCdim = ∞)
    - Implies that as $B \to \infty$, get $\inf_{\mathcal{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathcal{H}_{k,B}} L_{\mathcal{D}}(f) \to$ Bayes error if $\mathcal{D}$ has compact support
  - Can show universality via Stone-Weierstrass (more later), or Fourier properties
  - $\exp(x^\top y)$, $\exp(-\frac{1}{2\sigma^2}\|x - y\|^2)$, $\exp(-\frac{1}{\sigma}\|x - y\|)$ are universal on compact subsets of $\mathbb{R}^d$

Universal kernels

- What about that $L_S(f)$ or $\inf_{\|f\|_{\mathcal{H}_k}} L_{\mathcal{D}}(f)$ term?
- A continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}_k$ is dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, every $\varepsilon > 0$, there is an $f \in \mathcal{H}_k$
  with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \leq \varepsilon$
  - If $\mathcal{X}$ is a topological space *not* generated by a metric, there is no universal kernel (Steinwart/Christmann exercise 4.13)
  - Separates compact sets: if $X_1 \cap X_2 = \emptyset$ are compact subsets of $\mathcal{X}$,
    there's an $f \in \mathcal{H}_k$ with $f(x) > 0$ for $x \in X_1$, $f(x) < 0$ for $x \in X_2$ (so VCdim = ∞)
    - Implies that as $B \to \infty$, get $\inf_{\mathcal{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathcal{H}_{k,B}} L_{\mathcal{D}}(f) \to$ Bayes error if $\mathcal{D}$ has compact support
  - Can show universality via Stone-Weierstrass (more later), or Fourier properties
  - $\exp(x^\top y)$, $\exp(-\frac{1}{2\sigma^2}\|x - y\|^2)$, $\exp(-\frac{1}{\sigma}\|x - y\|)$ are universal on compact subsets of $\mathbb{R}^d$
  - Never true for finite-dimensional kernels

Approximation error

- Know that as $B \to \infty$, get $\inf_{\mathscr{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathscr{H}_{k,B}} L_{\mathscr{D}}(f) \to$ Bayes error

  for compactly supported $\mathscr{D}$ (can use broader notion of universality in general)

Approximation error

- Know that as $B \to \infty$, get $\inf_{\mathcal{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathcal{H}_{k,B}} L_{\mathcal{D}}(f) \to$ Bayes error for compactly supported $\mathcal{D}$ (can use broader notion of universality in general)
    - But the rate at which this happens depends on $\mathcal{D}$

Approximation error

- Know that as $B \to \infty$, get $\inf_{\mathscr{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathscr{H}_{k,B}} L_{\mathscr{D}}(f) \to$ Bayes error for compactly supported $\mathscr{D}$ (can use broader notion of universality in general)
  - But the rate at which this happens depends on $\mathscr{D}$
- Usually compare to the **regression function** $f_{\mathscr{D}}(x) = \mathbb{E}[y \mid x]$

Approximation error

- Know that as $B \to \infty$, get $\inf_{\mathscr{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathscr{H}_{k,B}} L_{\mathscr{D}}(f) \to$ Bayes error for compactly supported $\mathscr{D}$ (can use broader notion of universality in general)
  - But the rate at which this happens depends on $\mathscr{D}$
- Usually compare to the **regression function** $f_{\mathscr{D}}(x) = \mathbb{E}[y \mid x]$
  - If $f_{\mathscr{D}} \in \mathscr{H}_k$, called **well-specified**:

Approximation error

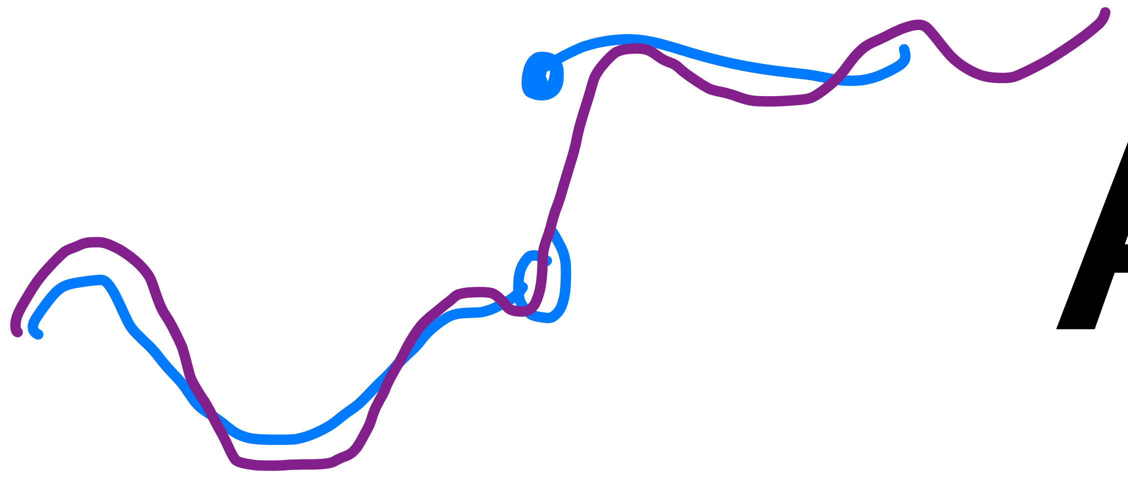
- Know that as $B \to \infty$, get $\inf_{\mathscr{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathscr{H}_{k,B}} L_{\mathscr{D}}(f) \to$ Bayes error for compactly supported $\mathscr{D}$ (can use broader notion of universality in general)
  - But the rate at which this happens depends on $\mathscr{D}$
- Usually compare to the **regression function** $f_{\mathscr{D}}(x) = \mathbb{E}[y \mid x]$
  - If $f_{\mathscr{D}} \in \mathscr{H}_k$, called **well-specified**:
    - Stability for $B = \|f_{\mathscr{D}}\|_{\mathscr{H}_k}$: $\inf_{\|f\|_{\mathscr{H}_k} \leq B} L_{\mathscr{D}}(f)$ = Bayes error, excess error $\leq \mathcal{O}(1/\sqrt{n})$

Approximation error

- Know that as $B \to \infty$, get $\inf_{\mathcal{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathcal{H}_{k,B}} L_{\mathcal{D}}(f) \to$ Bayes error for compactly supported $\mathcal{D}$ (can use broader notion of universality in general)
  - But the rate at which this happens depends on $\mathcal{D}$
- Usually compare to the **regression function** $f_{\mathcal{D}}(x) = \mathbb{E}[y \mid x]$
  - If $f_{\mathcal{D}} \in \mathcal{H}_k$, called **well-specified**:
    - Stability for $B = \|f_{\mathcal{D}}\|_{\mathcal{H}_k}$: $\inf_{\|f\|_{\mathcal{H}_k} \le B} L_{\mathcal{D}}(f)$ = Bayes error, excess error $\le \mathcal{O}(1/\sqrt{n})$
    - Better rates (minimax-optimal) with "range-space condition" if $f_{\mathcal{D}}$ is "nice" in $\mathcal{H}_k$

Approximation error

- Know that as $B \to \infty$, get $\inf_{\mathscr{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathscr{H}_{k,B}} L_{\mathscr{D}}(f) \to$ Bayes error for compactly supported $\mathscr{D}$ (can use broader notion of universality in general)
  - But the rate at which this happens depends on $\mathscr{D}$
- Usually compare to the **regression function** $f_{\mathscr{D}}(x) = \mathbb{E}[y \mid x]$
  - If $f_{\mathscr{D}} \in \mathscr{H}_k$, called **well-specified**:
    - Stability for $B = \|f_{\mathscr{D}}\|_{\mathscr{H}_k}$: $\inf_{\|f\|_{\mathscr{H}_k} \leq B} L_{\mathscr{D}}(f)$ = Bayes error, excess error $\leq \mathcal{O}(1/\sqrt{n})$
    - Better rates (minimax-optimal) with "range-space condition" if $f_{\mathscr{D}}$ is "nice" in $\mathscr{H}_k$
      - Pretty different style of analysis, based on $\|\hat{f} - f_{\mathscr{D}}\|_{\mathscr{H}_k}$



Approximation error

- Know that as $B \to \infty$, get $\inf_{\mathcal{H}_{k,B}} L_S(f) \to 0$, $\inf_{\mathcal{H}_{k,B}} L_{\mathcal{D}}(f) \to$ Bayes error for compactly supported $\mathcal{D}$ (can use broader notion of universality in general)
  - But the rate at which this happens depends on $\mathcal{D}$
- Usually compare to the **regression function** $f_{\mathcal{D}}(x) = \mathbb{E}[y \mid x]$
  - If $f_{\mathcal{D}} \in \mathcal{H}_k$, called **well-specified**:
    - Stability for $B = \|f_{\mathcal{D}}\|_{\mathcal{H}_k}$: $\inf_{\|f\|_{\mathcal{H}_k} \leq B} L_{\mathcal{D}}(f)$ = Bayes error, excess error $\leq \mathcal{O}(1/\sqrt{n})$
    - Better rates (minimax-optimal) with "range-space condition" if $f_{\mathcal{D}}$ is "nice" in $\mathcal{H}_k$
      - Pretty different style of analysis, based on $\|\hat{f} - f_{\mathcal{D}}\|_{\mathcal{H}_k}$
  - Misspecified case: more complicated analyses based on "approximation spaces"

Gaussian processes

- $f \sim \mathrm{GP}(m, k)$ is a **random function** $f : \mathscr{X} \to \mathbb{R}$ s.t., for any $x_1, \ldots, x_n$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathscr{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix} \right)$$

Gaussian processes

- $f \sim \mathrm{GP}(m, k)$ is a **random function** $f : \mathscr{X} \to \mathbb{R}$ s.t., for any $x_1, \ldots, x_n$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathscr{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix} \right)$$

- Mean function $m : \mathscr{X} \to \mathbb{R}$ can be any function; usually use 0

Gaussian processes

- $f \sim \mathrm{GP}(m, k)$ is a **random function** $f : \mathcal{X} \to \mathbb{R}$ s.t., for any $x_1, \ldots, x_n$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix} \right)$$

- Mean function $m : \mathcal{X} \to \mathbb{R}$ can be any function; usually use 0
  - will see that we can just shift everything by m so that this is WLOG

Gaussian processes

- $f \sim \mathrm{GP}(m, k)$ is a **random function** $f : \mathcal{X} \to \mathbb{R}$ s.t., for any $x_1, \ldots, x_n$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix} \right)$$

- Mean function $m : \mathcal{X} \to \mathbb{R}$ can be any function; usually use 0
    - will see that we can just shift everything by m so that this is WLOG
- Covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can be any psd function, i.e. any kernel

Gaussian processes

- $f \sim \mathrm{GP}(m, k)$ is a **random function** $f : \mathcal{X} \to \mathbb{R}$ s.t., for any $x_1, \ldots, x_n$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix} \right)$$

- Mean function $m : \mathcal{X} \to \mathbb{R}$ can be any function; usually use 0
  - will see that we can just shift everything by m so that this is WLOG
- Covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can be any psd function, i.e. any kernel

- Note: samples f are almost surely *not* in $\mathcal{H}_k$, for infinite-dim $\mathcal{H}_k$

Gaussian processes

- $f \sim \mathrm{GP}(m, k)$ is a **random function** $f : \mathcal{X} \to \mathbb{R}$ s.t., for any $x_1, \ldots, x_n$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix} \right)$$

- Mean function $m : \mathcal{X} \to \mathbb{R}$ can be any function; usually use 0
  - will see that we can just shift everything by m so that this is WLOG
- Covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can be any psd function, i.e. any kernel

- Note: samples f are almost surely *not* in $\mathcal{H}_k$, for infinite-dim $\mathcal{H}_k$
  - but they *are* almost surely in a "slightly bigger" RKHS
  - see e.g. Section 4 of Kanagawa et al. (2018)

Gaussian process regression

- Assume a **prior** $f \sim \mathrm{GP}(m, k)$

Gaussian process regression

- Assume a **prior** $f \sim \mathrm{GP}(m, k)$
- Assume **likelihood** of observations by $y_i \sim \mathcal{N}(f(x_i), \sigma^2)$

Gaussian process regression

- Assume a **prior** $f \sim \mathrm{GP}(m, k)$
- Assume **likelihood** of observations by $y_i \sim \mathcal{N}(f(x_i), \sigma^2)$
  - $\mathbb{E}[y_i] = \mathbb{E}[f(x_i)], \quad \mathrm{Cov}(y_i, y_j) = \mathrm{Cov}(f(x_i), f(x_j)) + \sigma^2 \delta_{ij}$

Gaussian process regression

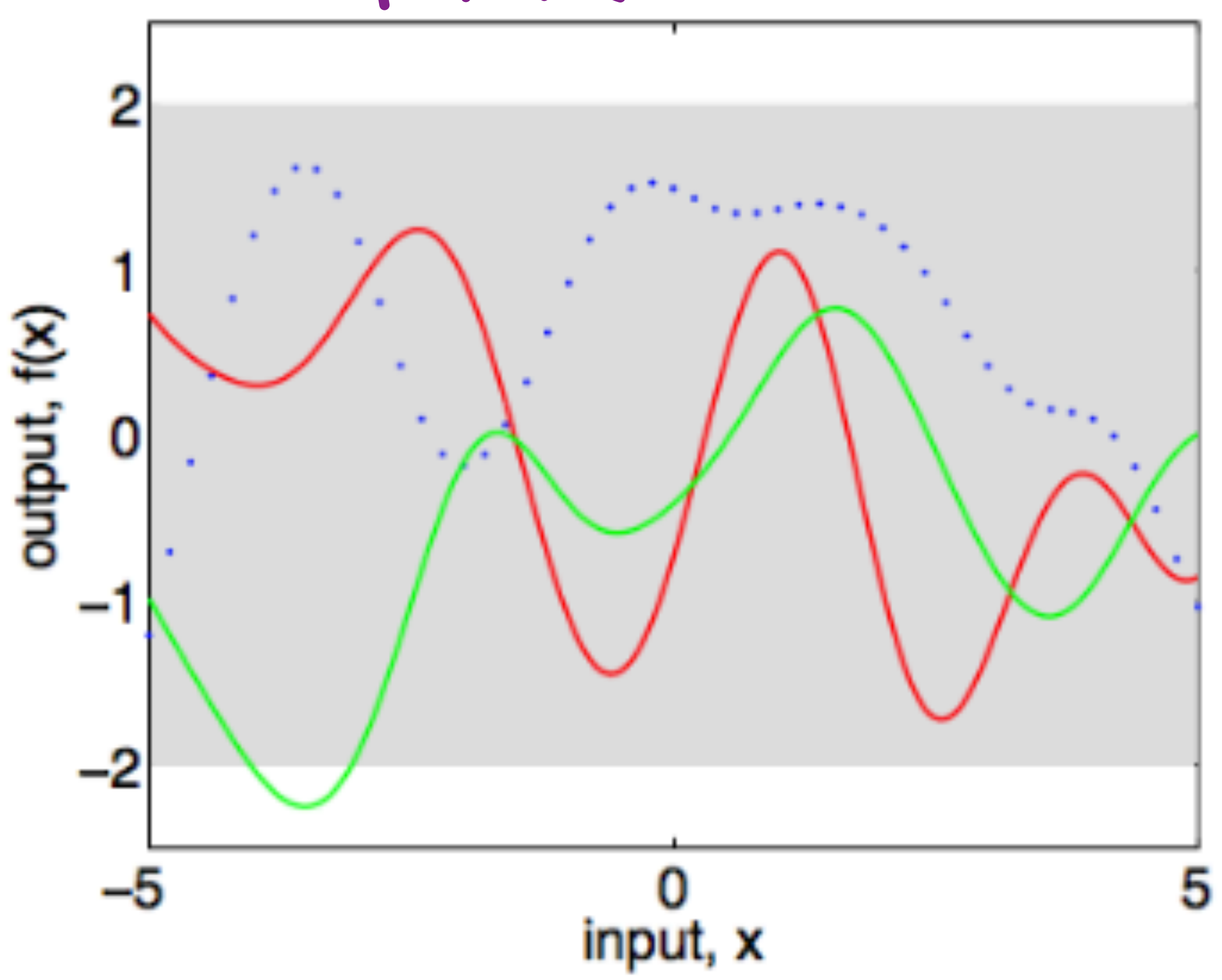
- Assume a **prior** $f \sim \mathrm{GP}(m, k)$
- Assume **likelihood** of observations by $y_i \sim \mathcal{N}(f(x_i), \sigma^2)$
  - $\mathbb{E}[y_i] = \mathbb{E}[f(x_i)], \quad \mathrm{Cov}(y_i, y_j) = \mathrm{Cov}(f(x_i), f(x_j)) + \sigma^2 \delta_{ij}$
- The **posterior** works out to be (via Kolmogorov Extension Theorem)

$$f \mid S \sim \mathrm{GP}\left(\left[x \mapsto y^\top (K_S + \sigma^2 I)^{-1} k_S(x)\right], \left[(x, x') \mapsto k(x, x') - k_S(x)^\top (K_S + \sigma^2 I)^{-1} k_S(x')\right]\right)$$

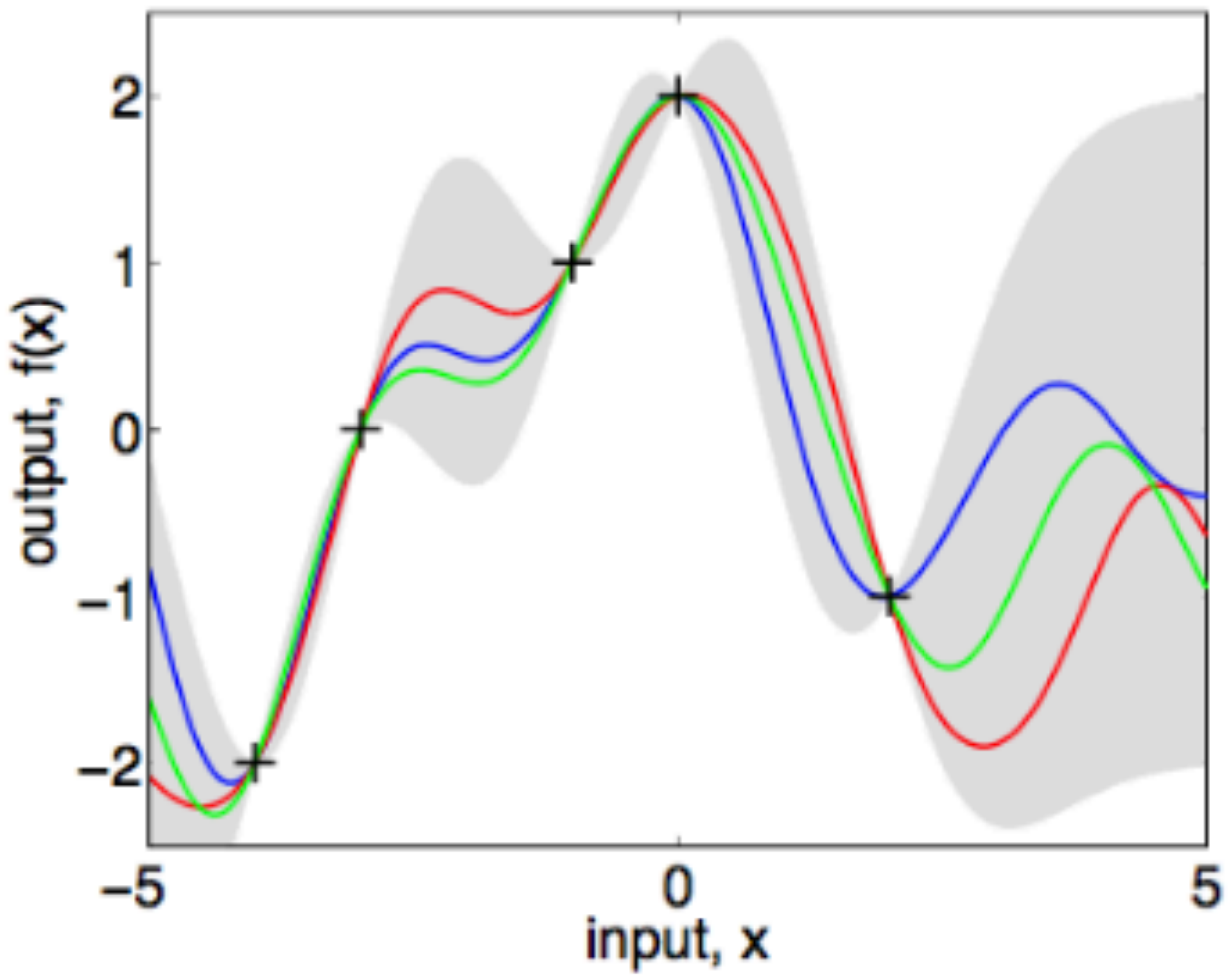
Gaussian process regression

- Assume a **prior** $f \sim \mathrm{GP}(m, k)$
- Assume **likelihood** of observations by $y_i \sim \mathcal{N}(f(x_i), \sigma^2)$
    - $\mathbb{E}[y_i] = \mathbb{E}[f(x_i)], \quad \mathrm{Cov}(y_i, y_j) = \mathrm{Cov}(f(x_i), f(x_j)) + \sigma^2 \delta_{ij}$
- The **posterior** works out to be (via Kolmogorov Extension Theorem)

$$f \mid S \sim \mathrm{GP}\left(\left[x \mapsto \underbrace{y^\top (K_S + \sigma^2 I)^{-1} k_S(x)}_{\text{KRR}}\right], \left[(x, x') \mapsto \underbrace{k(x, x') - k_S(x)^\top (K_S + \sigma^2 I)^{-1} k_S(x')}\right] \right)$$



(a), prior



(b), posterior

More Gaussian Processes

- GP regression: can get **posterior contraction rates**
    - Look like KRR analysis for the mean, plus posterior variance decreasing

- Understanding posterior variance can be very useful!
    - e.g. Bayesian optimization / active learning / bandits / …

- GP classifiers: usual choice corresponds to kernel logistic regression

More kernel resources

- Foundations: Berlinet and Thomas-Agnan, RKHSes in Probability and Stats (2004)
- Including more hardcore details: Steinwart and Christmann, SVMs (2008)
- Ridge regression analyses:
  - Smale and Zhou (2007) – fairly readable
  - Caponnetto and de Vito (2007) – minimax rate for “mostly”-well-specified, including regression with a kernel *output* as well
  - Steinwart et al. (2009) – minimax in Sobolev spaces with Matérn kernels (hard)

- Rasmussen and Williams, Gaussian Processes for Machine Learning (2006)

- Connections between kernels and GPs: Kanagawa et al. (2018)
- Mean embeddings: Muandet et al. (2016)
  - v. related to a lot of my research; there are slides in L16, but won’t get to them

$$K(x, x') = J_\theta(x)^T J_\theta(x')$$

$$\varphi : \mathcal{X} \to \mathbb{R}^p$$

$$K(x, x') = \mathcal{K}(\varphi(x), \varphi(x'))$$

(pause)

$$\varphi : \mathcal{X} \to \mathbb{R}^D$$

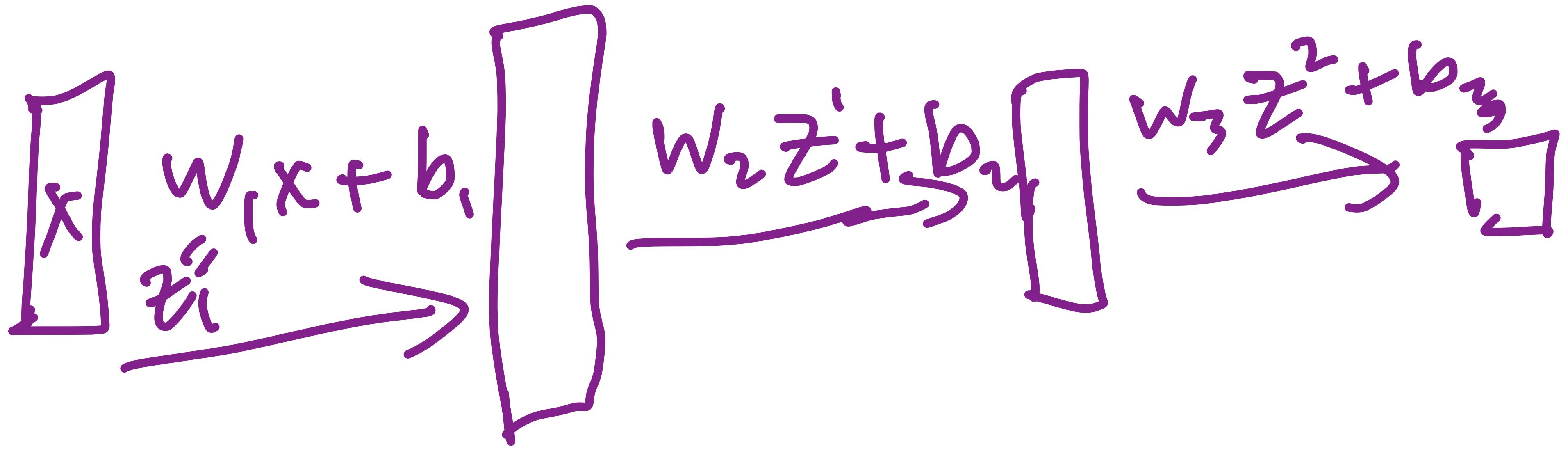
Gaussian or sth.

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets (“multilayer perceptrons”):

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
  - $f^{(0)}(x) = x$ $\qquad f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\qquad f(x) = f^{(L)}(x)$



Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
  - $f^{(0)}(x) = x$ $\qquad f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\qquad f(x) = f^{(L)}(x)$
  - $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\qquad b_\ell \in \mathbb{R}^{d'_\ell}$ $\qquad \sigma_\ell : \mathbb{R}^{d'_\ell} \to \mathbb{R}^{d_\ell}$ (usually $d'_\ell = d_\ell$)

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
  - $f^{(0)}(x) = x$ $\quad f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\quad f(x) = f^{(L)}(x)$
  - $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\quad b_\ell \in \mathbb{R}^{d'_\ell}$ $\quad \sigma_\ell : \mathbb{R}^{d'_\ell} \to \mathbb{R}^{d_\ell}$ (usually $d'_\ell = d_\ell$)
  - Can think of this as a directed, acyclic computation graph, organized in **layers**

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
  - $f^{(0)}(x) = x$ $\quad f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\quad f(x) = f^{(L)}(x)$
  - $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\quad b_\ell \in \mathbb{R}^{d'_\ell}$ $\quad \sigma_\ell : \mathbb{R}^{d'_\ell} \to \mathbb{R}^{d_\ell}$ (usually $d'_\ell = d_\ell$)
  - Can think of this as a directed, acyclic computation graph, organized in **layers**
  - Usually $\sigma_L(x) = x$; intermediate layers called **hidden layers**

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
    - $f^{(0)}(x) = x$ $\quad$ $f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\quad$ $f(x) = f^{(L)}(x)$
    - $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\quad$ $b_\ell \in \mathbb{R}^{d'_\ell}$ $\quad$ $\sigma_\ell : \mathbb{R}^{d'_\ell} \to \mathbb{R}^{d_\ell}$ (usually $d'_\ell = d_\ell$)
    - Can think of this as a directed, acyclic computation graph, organized in **layers**
    - Usually $\sigma_L(x) = x$; intermediate layers called **hidden layers**
    - Common choices for **activations** σ:

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
  - $f^{(0)}(x) = x$ $\qquad$ $f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\qquad$ $f(x) = f^{(L)}(x)$
  - $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\qquad$ $b_\ell \in \mathbb{R}^{d'_\ell}$ $\qquad$ $\sigma_\ell : \mathbb{R}^{d'_\ell} \to \mathbb{R}^{d_\ell}$ (usually $d'_\ell = d_\ell$)
  - Can think of this as a directed, acyclic computation graph, organized in **layers**
  - Usually $\sigma_L(x) = x$; intermediate layers called **hidden layers**
  - Common choices for **activations** σ:
    - Componentwise: $\mathrm{ReLU}(z) = \max\{z, 0\}, \mathrm{sigmoid}(z) = 1/(1 + \exp(-z))$

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
  - $f^{(0)}(x) = x$ $\quad f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\quad f(x) = f^{(L)}(x)$
  - $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\quad b_\ell \in \mathbb{R}^{d'_\ell}$ $\quad \sigma_\ell : \mathbb{R}^{d'_\ell} \to \mathbb{R}^{d_\ell}$ (usually $d'_\ell = d_\ell$)
  - Can think of this as a directed, acyclic computation graph, organized in **layers**
  - Usually $\sigma_L(x) = x$; intermediate layers called **hidden layers**
  - Common choices for **activations** σ:
    - Componentwise: $\mathrm{ReLU}(z) = \max\{z, 0\}, \mathrm{sigmoid}(z) = 1/(1 + \exp(-z))$
    - $\mathrm{softmax}(z)_i = \exp(z_i)/\sum_j \exp(z_j)$, max pooling, attention, …

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
  - $f^{(0)}(x) = x$ $\quad f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\quad f(x) = f^{(L)}(x)$
  - $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\quad b_\ell \in \mathbb{R}^{d'_\ell}$ $\quad \sigma_\ell : \mathbb{R}^{d'_\ell} \to \mathbb{R}^{d_\ell}$ (usually $d'_\ell = d_\ell$)
  - Can think of this as a directed, acyclic computation graph, organized in **layers**
  - Usually $\sigma_L(x) = x$; intermediate layers called **hidden layers**
  - Common choices for **activations** σ:
    - Componentwise: $\mathrm{ReLU}(z) = \max\{z, 0\}, \mathrm{sigmoid}(z) = 1/(1 + \exp(-z))$
    - $\mathrm{softmax}(z)_i = \exp(z_i)/\sum_j \exp(z_j)$, max pooling, attention, …
  - Usually train via SGD, but it's **non-convex**: in general, possibility of local minima

Deep learning

- Mostly assuming **fully-connected**, **feedforward** nets ("multilayer perceptrons"):
  - $f^{(0)}(x) = x$ $\quad f^{(\ell)}(x) = \sigma_\ell(W_\ell f^{(\ell-1)}(x) + b_\ell)$ $\quad f(x) = f^{(L)}(x)$
  - $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\quad b_\ell \in \mathbb{R}^{d'_\ell}$ $\quad \sigma_\ell : \mathbb{R}^{d'_\ell} \to \mathbb{R}^{d_\ell}$ (usually $d'_\ell = d_\ell$)
  - Can think of this as a directed, acyclic computation graph, organized in **layers**
  - Usually $\sigma_L(x) = x$; intermediate layers called **hidden layers**
  - Common choices for **activations** σ:
    - Componentwise: $\mathrm{ReLU}(z) = \max\{z, 0\}, \mathrm{sigmoid}(z) = 1/(1 + \exp(-z))$
    - $\mathrm{softmax}(z)_i = \exp(z_i) / \sum_j \exp(z_j)$, max pooling, attention, …
  - Usually train via SGD, but it's **non-convex**: in general, possibility of local minima
    - ERM is NP-hard, even with 1 ReLU, *even for square loss* (Goel et al. ITCS 2021)

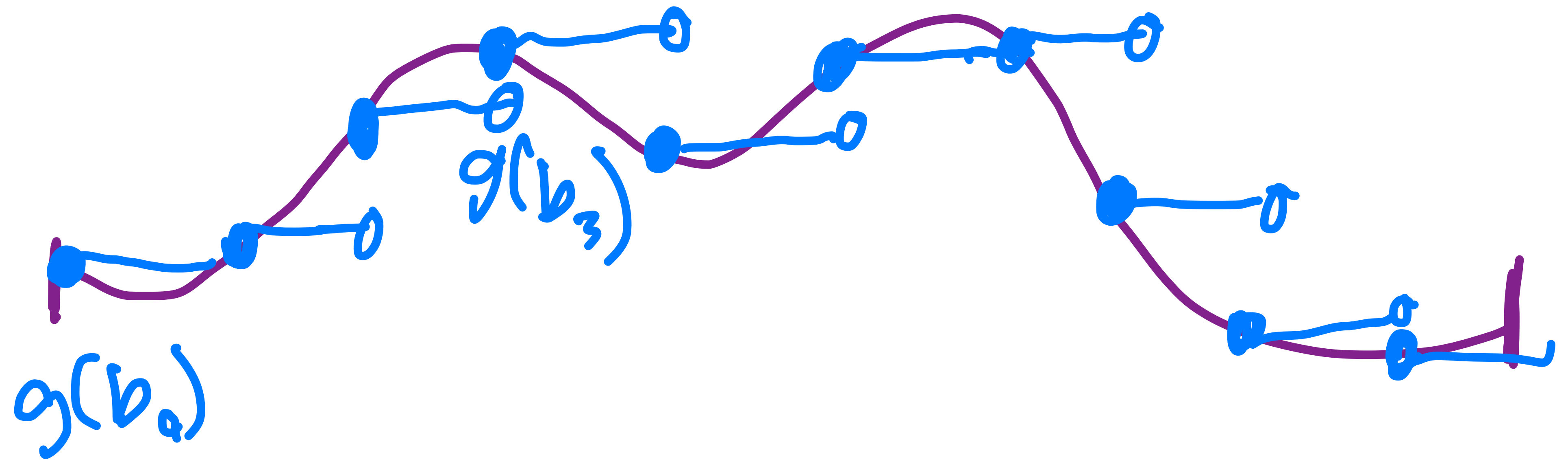
Universal approximation in $\mathbb{R}$

Theorem: Let $g : \mathbb{R} \to \mathbb{R}$ be ρ-Lipschitz. For any $\varepsilon > 0$, there is a two-layer network f with $m := \lceil \frac{\rho}{\varepsilon} \rceil$ hidden nodes, $\sigma_1(z) = \mathbb{I}(z \geq 0)$, with $\sup_{x \in [0,1]} |f(x) - g(x)| \leq \varepsilon$.

Universal approximation in $\mathbb{R}$

Theorem: Let $g : \mathbb{R} \to \mathbb{R}$ be ρ-Lipschitz. For any $\varepsilon > 0$, there is a two-layer network f with $m := \lceil \frac{\rho}{\varepsilon} \rceil$ hidden nodes, $\sigma_1(z) = \mathbb{I}(z \geq 0)$, with $\sup_{x \in [0,1]} |f(x) - g(x)| \leq \varepsilon$.

$$b_i = \frac{i\varepsilon}{\rho} \qquad a_0 = g(0) \qquad a_i = g(b_i) - g(b_{i-1}) \qquad f(x) = \sum_{i=0}^{m-1} a_i \mathbb{I}(x_i \geq b_i)$$



Universal approximation in $\mathbb{R}$

Theorem: Let $g : \mathbb{R} \to \mathbb{R}$ be ρ-Lipschitz. For any $\varepsilon > 0$, there is a two-layer network f with $m := \lceil \frac{\rho}{\varepsilon} \rceil$ hidden nodes, $\sigma_1(z) = \mathbb{I}(z \geq 0)$, with $\sup_{x \in [0,1]} |f(x) - g(x)| \leq \varepsilon$.

$$b_i = \frac{i\varepsilon}{\rho} \qquad a_0 = g(0) \qquad a_i = g(b_i) - g(b_{i-1}) \qquad f(x) = \sum_{i=0}^{m-1} a_i \mathbb{I}(x_i \geq b_i)$$

$$|\, g(x) - f(x) \,|$$

Universal approximation in $\mathbb{R}$

Theorem: Let $g : \mathbb{R} \to \mathbb{R}$ be ρ-Lipschitz. For any $\varepsilon > 0$, there is a two-layer network f with $m := \lceil \frac{\rho}{\varepsilon} \rceil$ hidden nodes, $\sigma_1(z) = \mathbb{I}(z \geq 0)$, with $\sup_{x \in [0,1]} |f(x) - g(x)| \leq \varepsilon$.

$$b_i = \frac{i\varepsilon}{\rho} \qquad a_0 = g(0) \qquad a_i = g(b_i) - g(b_{i-1}) \qquad f(x) = \sum_{i=0}^{m-1} a_i \mathbb{I}(x_i \geq b_i)$$

$$k = \max\{k : b_k \leq x\}$$

$$|g(x) - f(x)| \leq |g(x) - g(b_k)| + \underbrace{|g(b_k) - f(b_k)|}_{0} + \underbrace{|f(b_k) - f(x)|}_{0}$$

Universal approximation in $\mathbb{R}$

Theorem: Let $g : \mathbb{R} \to \mathbb{R}$ be ρ-Lipschitz. For any $\varepsilon > 0$, there is a two-layer network f with $m := \lceil \frac{\rho}{\varepsilon} \rceil$ hidden nodes, $\sigma_1(z) = \mathbb{I}(z \geq 0)$, with $\sup_{x \in [0,1]} |f(x) - g(x)| \leq \varepsilon$.

$$b_i = \frac{i\varepsilon}{\rho} \qquad a_0 = g(0) \qquad a_i = g(b_i) - g(b_{i-1}) \qquad f(x) = \sum_{i=0}^{m-1} a_i \mathbb{I}(x_i \geq b_i)$$

$$k = \max\{k : b_k \leq x\}$$

$$|g(x) - f(x)| \leq |g(x) - g(b_k)| + |g(b_k) - f(b_k)| + |f(b_k) - f(x)|$$

$$\leq \rho |x - b_k|$$

Universal approximation in $\mathbb{R}$

Theorem: Let $g : \mathbb{R} \to \mathbb{R}$ be ρ-Lipschitz. For any $\varepsilon > 0$, there is a two-layer network f with $m := \lceil \frac{\rho}{\varepsilon} \rceil$ hidden nodes, $\sigma_1(z) = \mathbb{I}(z \geq 0)$, with $\sup_{x \in [0,1]} |f(x) - g(x)| \leq \varepsilon$.

$$b_i = \frac{i\varepsilon}{\rho} \qquad a_0 = g(0) \qquad a_i = g(b_i) - g(b_{i-1}) \qquad f(x) = \sum_{i=0}^{m-1} a_i \mathbb{I}(x_i \geq b_i)$$

$$k = \max\{k : b_k \leq x\}$$

$$\begin{aligned} |\,g(x) - f(x)\,| &\leq |\,g(x) - g(b_k)\,| + |\,g(b_k) - f(b_k)\,| + |\,f(b_k) - f(x)\,| \\ &\leq \rho\,|\,x - b_k\,| \\ &\leq \rho\frac{\varepsilon}{\rho} = \varepsilon \end{aligned}$$

Universal approximation in $\mathbb{R}$

Theorem: Let $g : \mathbb{R} \to \mathbb{R}$ be ρ-Lipschitz. For any $\varepsilon > 0$, there is a two-layer network f with $m := \lceil \frac{\rho}{\varepsilon} \rceil$ hidden nodes, $\sigma_1(z) = \mathbb{1}(z \geq 0)$, with $\sup_{x \in [0,1]} |f(x) - g(x)| \leq \varepsilon$.

$$b_i = \frac{i\varepsilon}{\rho} \qquad a_0 = g(0) \qquad a_i = g(b_i) - g(b_{i-1}) \qquad f(x) = \sum_{i=0}^{m-1} a_i \mathbb{1}(x_i \geq b_i)$$

$$k = \max\{k : b_k \leq x\}$$

$$\begin{aligned}
|g(x) - f(x)| &\leq |g(x) - g(b_k)| + |g(b_k) - f(b_k)| + |f(b_k) - f(x)| \\
&\leq \rho |x - b_k| \\
&\leq \rho \frac{\varepsilon}{\rho} = \varepsilon
\end{aligned}$$

Can do better by depending on *total variation* of g

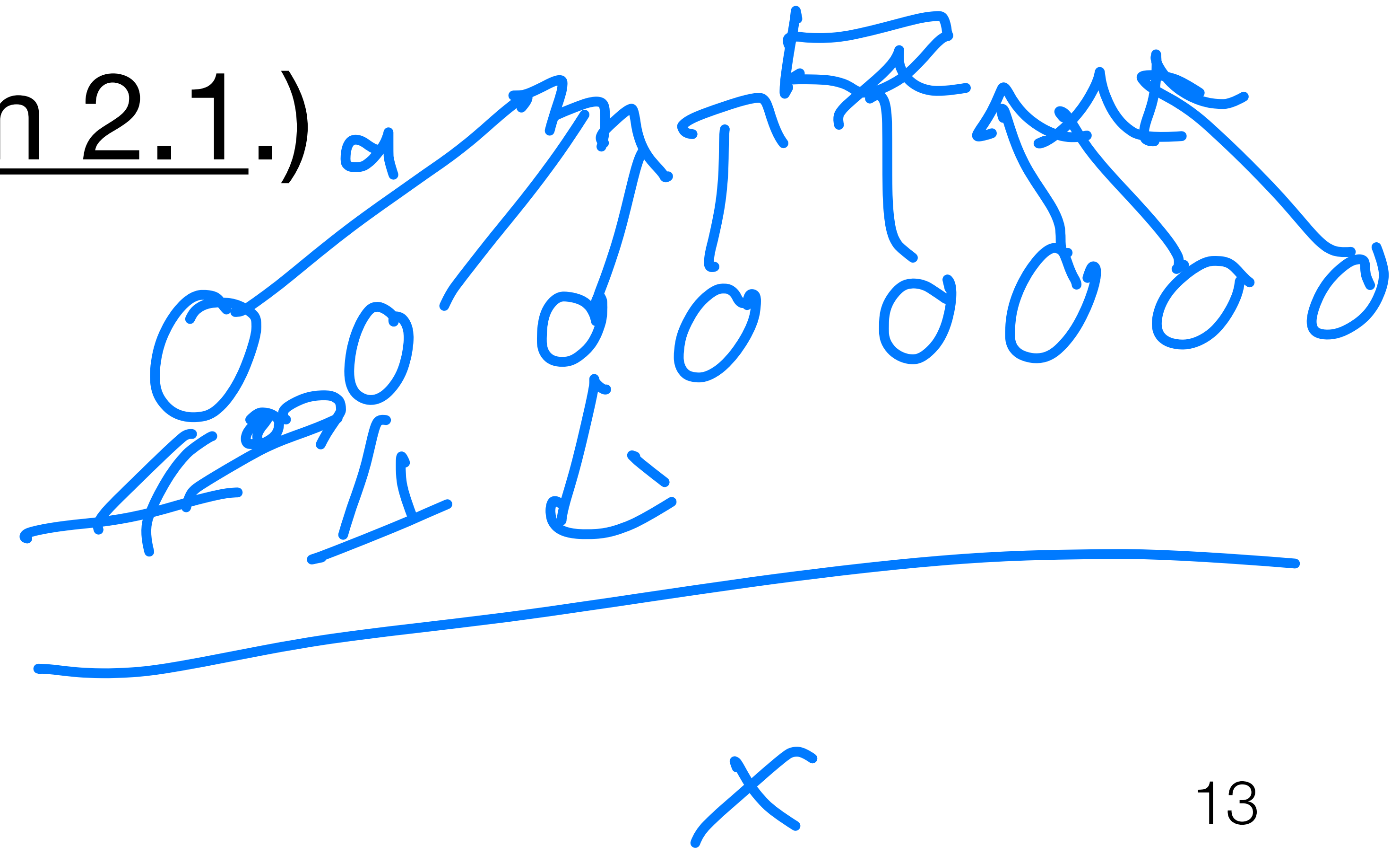
Universal approximation in $\mathbb{R}^d$

Theorem: Let $g : \mathbb{R}^d \to \mathbb{R}$ be continuous. For any $\varepsilon > 0$, choose $\delta > 0$ so that $\|x - x'\|_\infty \leq \delta$ implies $|g(x) - g(x')| \leq \varepsilon$. Then there is a three-layer ReLU network f with $\Omega\left(\frac{1}{\delta^d}\right)$ nodes satisfying $\int_{[0,1]^d} |f(x) - g(x)| \mathrm{d}x \leq 2\varepsilon$.

Universal approximation in $\mathbb{R}^d$

Theorem: Let $g : \mathbb{R}^d \to \mathbb{R}$ be continuous. For any $\varepsilon > 0$, choose $\delta > 0$ so that $\|x - x'\|_\infty \le \delta$ implies $|g(x) - g(x')| \le \varepsilon$. Then there is a three-layer ReLU network f with $\Omega\left(\frac{1}{\delta^d}\right)$ nodes satisfying $\int_{[0,1]^d} |f(x) - g(x)| \mathrm{d}x \le 2\varepsilon$.

Proof approximates continuous g by piecewise-constant h,
then uses a two-layer ReLU net to check if x is in each piece, roughly like in 1d.
(Telgarsky's Theorem 2.1.)



Universal approximation in $\mathbb{R}^d$, one hidden layer

Stone-Weierstrass Theorem: Let $\mathscr{F}$ be a set of functions such that

1. Each $f \in \mathscr{F}$ is continuous.
2. For each x, there is at least one $f \in \mathscr{F}$ with $f(x) \neq 0$.
3. Separates points: for each $x \neq x'$, there is at least one $f \in \mathscr{F}$ with $f(x) \neq f(x')$.
4. $\mathscr{F}$ is an algebra: for $f, g \in \mathscr{F}$, $\quad \alpha f + g \in \mathscr{F}$ and $fg = (x \mapsto f(x)g(x)) \in \mathscr{F}$.

Then $\mathscr{F}$ is dense in $C(\mathcal{X})$ w.r.t. $\|\cdot\|_\infty$.

Universal approximation in $\mathbb{R}^d$, one hidden layer

Stone-Weierstrass Theorem: Let $\mathcal{F}$ be a set of functions such that

1. Each $f \in \mathcal{F}$ is continuous.
2. For each x, there is at least one $f \in \mathcal{F}$ with $f(x) \neq 0$.
3. Separates points: for each $x \neq x'$, there is at least one $f \in \mathcal{F}$ with $f(x) \neq f(x')$.
4. $\mathcal{F}$ is an algebra: for $f, g \in \mathcal{F}$, $\quad \alpha f + g \in \mathcal{F}$ and $fg = (x \mapsto f(x)g(x)) \in \mathcal{F}$.

Conditions hold for $\sigma_1 = \exp$, $\sigma_2 = \mathrm{Id}$, so that $\mathcal{F}_{\exp} = \{x \mapsto \sum_{i=1}^{m} a_i \exp(w_i^\top x)\}$

Universal approximation in $\mathbb{R}^d$, one hidden layer

Stone-Weierstrass Theorem: Let $\mathcal{F}$ be a set of functions such that

1. Each $f \in \mathcal{F}$ is continuous.
2. For each x, there is at least one $f \in \mathcal{F}$ with $f(x) \neq 0$.
3. Separates points: for each $x \neq x'$, there is at least one $f \in \mathcal{F}$ with $f(x) \neq f(x')$.
4. $\mathcal{F}$ is an algebra: for $f, g \in \mathcal{F}$, $\quad \alpha f + g \in \mathcal{F} \quad$ and $\quad fg = (x \mapsto f(x)g(x)) \in \mathcal{F}$.

Conditions hold for $\sigma_1 = \exp$, $\sigma_2 = \mathrm{Id}$, so that $\mathcal{F}_{\exp} = \{x \mapsto \sum_{i=1}^{m} a_i \exp(w_i^\top x)\}$

If $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous, $\lim_{z \to -\infty} \sigma(z) = 0$, $\lim_{z \to -\infty} \sigma(z) = 1$, works too:

Approximate g by $h \in \mathcal{F}_{\exp}$ with $\frac{\varepsilon}{2}$ error, and replace each $\exp$ with a 1d σ-based net

Universal approximation in $\mathbb{R}^d$, one hidden layer

Stone-Weierstrass Theorem: Let $\mathscr{F}$ be a set of functions such that

1. Each $f \in \mathscr{F}$ is continuous.
2. For each x, there is at least one $f \in \mathscr{F}$ with $f(x) \neq 0$.
3. Separates points: for each $x \neq x'$, there is at least one $f \in \mathscr{F}$ with $f(x) \neq f(x')$.
4. $\mathscr{F}$ is an algebra: for $f, g \in \mathscr{F}$, $\quad \alpha f + g \in \mathscr{F}$ and $fg = (x \mapsto f(x)g(x)) \in \mathscr{F}$.

Conditions hold for $\sigma_1 = \exp$, $\sigma_2 = \mathrm{Id}$, so that $\mathscr{F}_{\exp} = \{x \mapsto \sum_{i=1}^{m} a_i \exp(w_i^\top x)\}$

If $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous, $\lim_{z \to -\infty} \sigma(z) = 0$, $\lim_{z \to -\infty} \sigma(z) = 1$, works too:

Approximate g by $h \in \mathscr{F}_{\exp}$ with $\frac{\varepsilon}{2}$ error, and replace each $\exp$ with a 1d σ-based net

Generally: universal approximator iff σ is **not** a polynomial

Circuit complexity

SSBD chapter 20:

- 2 layer nets with sign activations can represent all functions $\{\pm 1\}^d \to \{\pm 1\}$

Circuit complexity

SSBD chapter 20:

- 2 layer nets with sign activations can represent all functions $\{\pm 1\}^d \to \{\pm 1\}$
  - (remember that computers always represent things as $\{0,1\}^d \ldots$)

Circuit complexity

SSBD chapter 20:

- 2 layer nets with sign activations can represent all functions $\{\pm 1\}^d \to \{\pm 1\}$
  - (remember that computers always represent things as $\{0,1\}^d$…)
- …but, it takes exponential width to do that

Circuit complexity

SSBD chapter 20:

- 2 layer nets with sign activations can represent all functions $\{\pm 1\}^d \to \{\pm 1\}$
    - (remember that computers always represent things as $\{0,1\}^d$…)
- …but, it takes exponential width to do that
- …but, there's a network of size $\mathcal{O}(T^2)$ that can implement all boolean functions that can be computed in maximum runtime T

Limits of universal approximation

- Curse of dimensionality: usually requires # of units exponential in dimension
  - Also usually requires exponential norm of weights

- Doesn't say anything about whether ERM finds a good network, just that one exists
  - Let alone anything about whether (S)GD finds it