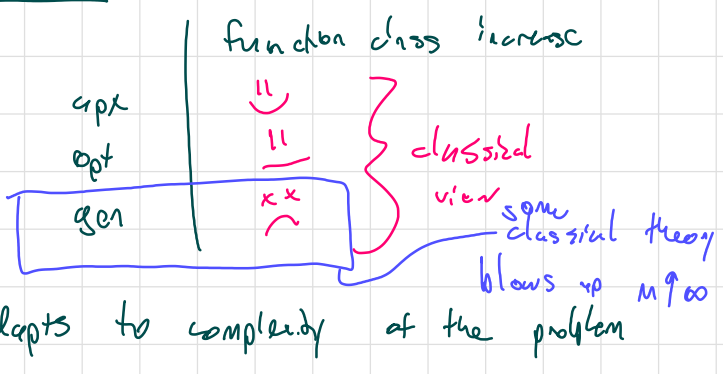
Lec 7: "kernel" part of NTK; start of deep/shallow separation.

Announcements

* Hw1 due 9/22 (no late submissions!)
* NTK notes up; section 5 tonight.
* Loomis 144 Thursday only; short OH today.

Question.



fix: DL adapts to complexity of the problem

Recap

$$f(x;W) := \frac{1}{\sqrt{m}} \sum_{j=1}^{m} a_j \sigma(w_j^T x)$$

$$f_0(x;W) := f(x;W_0) + \langle \nabla f(x;W_0), W - W_0 \rangle$$

$$|f(x;W) - f_0(x;W)| \le \begin{cases} O\left(\frac{\|W-W_0\|_2^2}{\sqrt{m}}\right) & \sigma \text{ smooth} \\ O\left(\frac{\|W-W_0\|^{4/3}}{m^{1/2}}\right) & \sigma \text{ ReLU} \\ = O\left(\left(\frac{\|W-W_0\|_2^4}{\sqrt{m}}\right)^{1/3}\right) & \end{cases}$$

Today: "kernel story".

$$k_m(x,x') := \langle \nabla f(x;W_0), \nabla f(x';W_0) \rangle \quad \text{(ReLU)}$$

$$= \left\langle \begin{bmatrix} a_1 \mathbb{1}[w_{0,1}^T x \ge 0] x/\sqrt{m} \\ \vdots \\ a_m \mathbb{1}[w_{0,m}^T x \ge 0] x/\sqrt{m} \end{bmatrix}, \begin{bmatrix} — & x' & — \\ — & x' & — \end{bmatrix} \right\rangle$$

$$= \left(\frac{1}{\sqrt{m}}\right)^2 \sum_{j=1}^{m} (x^T x') a_j^2 \mathbb{1}[w_{0,j}^T x \ge 0] \cdot \mathbb{1}[w_{0,j}^T x' \ge 0]$$

$$= (x^T x') \left[\frac{1}{m} \sum_{j=1}^{m} \mathbb{1}[w_{0,j}^T x \ge 0] \cdot \mathbb{1}[w_{0,j}^T x' \ge 0] \right]$$

m summands are iid.

$$\xrightarrow[m \to \infty]{\text{SLLN}} (x^T x') \underbrace{\mathbb{E}_w \mathbb{1}[w^T x \ge 0] \cdot \mathbb{1}[w^T x' \ge 0]}_{k(x,x')} \qquad (w \sim \sigma')$$

General activation

$$(x^T x') \mathbb{E}_w \sigma'(w^T x) \sigma'(w^T x')$$

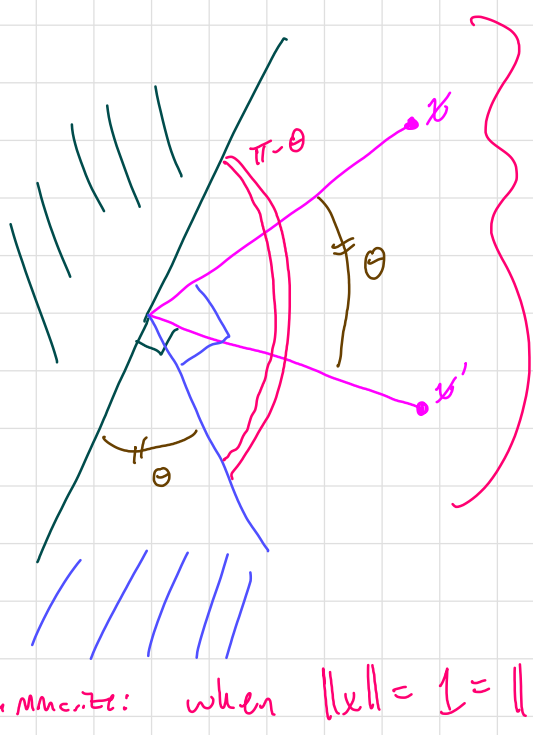
kernel for ReLU

$$\mathbb{E}_w \mathbb{1}[w^T x \ge 0] \cdot \mathbb{1}[w^T x' \ge 0]$$

$$= \Pr_w [w^T x \ge 0 \wedge w^T x' \ge 0]$$

$$= \Pr_w \left[\frac{w^T}{\|w\|} x \ge 0 \wedge \frac{w^T}{\|w\|} x' \ge 0 \right]$$

Note: norm of w doesn't matter;



$$= \frac{\pi - \theta}{2\pi} = \frac{\pi - \arccos(x^T x')}{2\pi}$$

Summary: when $\|x\| = 1 = \|x'\|$

$$k_{relu}(x,x') = (x^T x') \left[\frac{\pi - \arccos(x^T x')}{2\pi} \right].$$

Remark (why kernel):

Things people do w/ kernel view:

* Evolution of weights can be modeled as "Gaussian Process" (cf. NTK paper (Clement-Jacot-Hongler))
* acos kernel universal approximator
* Minimum eigenvalue of Gram matrix can lead to optimization rates.

Remark Inner product kernels have many special properties which NTK inherits.

Remark. Multi-layer kernel (not well understood) (power vs shallow not well understood)

$$\langle \nabla f(x;\vec{W}(0)), \nabla f(x';\vec{W}(0)) \rangle$$

$$= \sum_{j=1}^{L} \langle \nabla_j f(x;\vec{W}(0)), \nabla_j f(x';\vec{W}(0)) \rangle. \quad //$$

Remark. Taylor expansion at 0:

$$\langle \nabla f(x;0), \nabla f(x';0) \rangle$$

$$= \left\langle \begin{bmatrix} a_1 \frac{1}{\sqrt{m}} x \sigma'(0) \\ \vdots \\ a_m \frac{1}{\sqrt{m}} x \sigma'(0) \end{bmatrix}, \begin{bmatrix} a_1 \frac{1}{\sqrt{m}} x' \sigma'(0) \\ \vdots \end{bmatrix} \right\rangle$$

using deterministic fixed $\sigma'(0)$ "semi gradient"

$$= x^T x' \frac{1}{m} \sum_j a_j^2 \sigma'(0)\sigma'(0) = x^T x' \sigma'(0)^2.$$

Bach's phrase:

$$\sigma(\sigma(x) - \sigma(-x)) - \sigma(-x)$$

Remark. Can prove acos kernel gives a universal approximator. (Nodes used a bias)

$\Rightarrow$ "closes the loop"

with large m: $f_0(\cdot;W) \approx f(\cdot;W)$

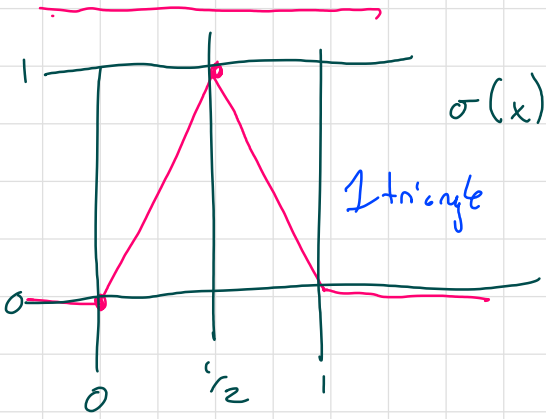
contains g

Deep vs shallow (impractical)

① "Δ" quickly increase complexity with depth; only deep vs shallow.

② $\exists$ small deep network which can't be approximated by subexponentially sized shallow networks.

③ Can use Δ to build: polynomials, Taylor apx ...

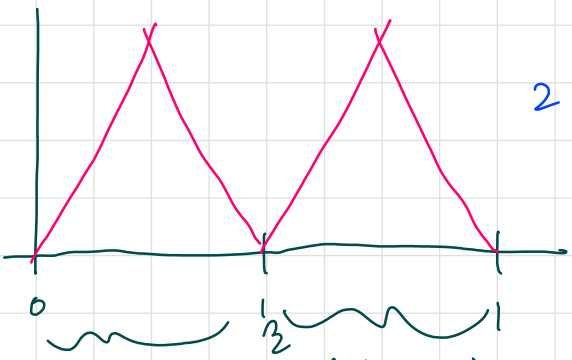


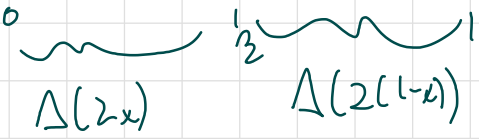
$$\sigma(x) = 2\sigma(x) - 4\sigma(x-\tfrac{1}{2}) + 2\sigma(x-1)$$

$$= \begin{cases} 2x & x \in [0, \tfrac{1}{2}] \\ 2(1-x) & x \in (\tfrac{1}{2}, 1] \\ 0 & \text{otherwise} \end{cases}$$

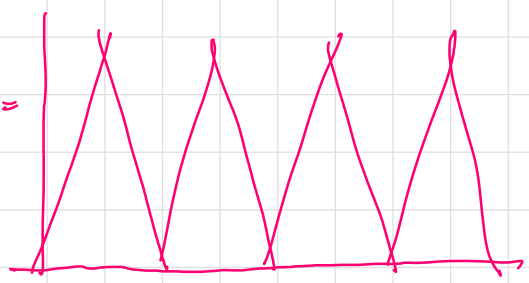
Credit? (dynamical systems? Bengio et al. "folding"? turing completeness of RNNs? ...)

$\Delta \circ \Delta(x) =$





$\Delta \circ \Delta \circ \Delta(x) =$



$\Delta^k(x) = (\Delta \circ \cdots \circ \Delta)(x) =$ 2^{k-1} triangles.

k copies

Point: depth allows complexity to increase exponentially fast.

Remark (impractical).

No evidence this relates to power of deep networks trained by gradient descent.

Office hours

$\mathcal{H}$

$$\left\{ x \mapsto \sum_{j=1}^{N} a_j k(x, x_j) : \begin{array}{l} N \geq 0, \\ a_j \in \mathbb{R} \\ x_j \in X \end{array} \right\}$$

norm: $\sum_{i,j} a_i a_j k(x_i, x_j)$

$$\mathcal{H}_B := \{ f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B \}$$

Assn: continuous g $\exists h \in \mathcal{H}$ s.t. $\|g - f\| \leq \varepsilon$.

$|f - f_0| \xrightarrow[n \to \infty]{} 0.$

Given cont g $\varepsilon > 0$,

Pick $h \in \mathcal{H}$, $\|h\|_{\mathcal{H}} < \infty$

$|f - f_0| = \frac{\|w - w_0\|_2^2}{\sqrt{n}} \sim \|h\|_{\mathcal{H}}$

$\frac{d}{dr} f(rx) = x f'(rx)$

$f(z) = z^2 \quad f'(z) = 2z$