This talk is about

On Lazy training in Diff. Prg.
by Chizat - Oyallon - Bach

three big things today

1) Any parametric model trained by a gradient flow has an induced function evolution governed by a time-dependent kernel. This is exact.

2) If the tangent kernel

stays nearly constant, training is a kernel gradient descent in a fixed RKHS.

3) Lazy training is a phenomenon where the NTK exists, yet in many practical settings, good performance correlates with leaving the NTK regime.

Setup: Let $f_\theta : \mathbb{R}^d \to \mathbb{R}$ (diff.)

$\theta \in \mathbb{R}^p$

Given $(x_i, y_i)_{i=1}^n$, define

$$L(\theta) = \frac{1}{2} \sum_{i=1}^{n} (f_\theta(x_i) - y_i)^2$$

G. flow is

$$\dot{\theta} = -\nabla_\theta L(\theta(t))$$

Notation:

$$f_t(x) := f_{\theta(t)}(x)$$

$$\vec{f}(t) := (f_t(x_1) \dots f_t(x_n))^T \in \mathbb{R}^n$$

$$\vec{y} := (y_1, \dots, y_n)^T$$

Calculus time

$$\dot{\theta} = -\nabla_\theta \mathcal{L} = -\sum_{i=1}^{n} (f_\theta(x_i) - y_i) \nabla_\theta f_\theta(x_i)$$

$$\partial_t f_t = \nabla_\theta f_{\theta(t)}(x) \cdot \dot{\theta}(t)$$

$$= -\sum_{i=1}^{n} \langle \nabla_\theta f_t(x), \nabla_\theta f(x_i) \rangle (f_t(x_i) - y_i)$$

$$= -\sum_{i=1}^{n} \underbrace{\Theta_{\theta(t)}(x, x_i)}_{\text{Universal Kernel}} (f_t(x_i) - y_i)$$

Define

$$\Theta(t) \in \mathbb{R}^{n \times n}$$

$$\Theta(t)_{ij} := \Theta_{\theta(t+1)}(x_i, x_j)$$

Then

$$\dot{\vec{f}} = -\Theta(t)(\vec{f}(t) - \vec{y})$$

Let $r(t) = \vec{f}(t) - \vec{y}$

Then

$$\dot{r}(t) = -\Theta(t) r(t)$$

Q: How does Θ behave when we use neural nets?

Q2: What would be the easiest Θ to use with? (A: constant & P.D.)

Assume $\Theta \equiv \Theta_0$ constant, P.D.

$$\dot{r}(t) = \Theta_0 r(t)$$

$$r(0) = \vec{f}(0) - \vec{y}$$

$$r(t) = e^{-t\Theta_0} r(0)$$

$$\Rightarrow \vec{f}(t) = \vec{y} + e^{-t\Theta_0}(\vec{f}(0) - \vec{y})$$

$$\longrightarrow y \text{ as } t \to \infty.$$

Perfect. Now time for a kernel trick

Define

$$K_0(x) := \left(\Theta_{\theta_0}(x, x_1), \dots, \Theta_{\theta_0}(x, x_n)\right)^T$$

(Still assuming $\theta(0) = \theta(t)$ ...)

test point ODE (just recursive Universal kernel eq)

$$\partial_t f_t(x) = -K_0(x)^T r(t)$$

$$= -K_0(x)^T e^{-t\Theta_0} r(0)$$

$$\rightarrow f_t(x) = f_0(x) - K_0(x)^T \int_0^t e^{-s\Theta_0} ds\, r(0)$$

$$= f_0(x) - K_0(x)^T \Theta_0^{-1}(I - e^{-t\Theta_0}) r(0)$$

$$\longrightarrow \quad t \to \infty$$

$$= f_0(x) - K_0(x)^T \Theta_0^{-1}(\vec{f}(0) - \vec{y})$$

Imagine you had $f_0(x) \equiv 0$

$$f_\infty(x) = K_0(x)^T \Theta_0^{-1} y$$

the representer form of kernel interpolation on the training set.

End of (2)

This is a complete theory! If only $\Theta(t)$ constant..

Start (3):

Let's look at linearization the typical way.

$$f_\theta(x) = f_{\theta_0}(x) + \nabla_\theta f_{\theta_0}(x) \cdot (\theta - \theta_0) + R_2(\theta; x)$$

$\exists \; s \in (0,1)$

$$R_2(\theta; x) = \frac{1}{2}(\theta - \theta_0^T) \left[\nabla_\theta^2 f_{\theta_0 + s(\theta - \theta_0)}(x) \right] (\theta - \theta_0)$$

Lang teaching says

$\exists$ a region parameter

$$\|\theta(t) - \theta_0\| \to 0$$

such

$$f_\theta^{lin}(x) := f_{\theta_0}(x) + \nabla_\theta f_{\theta_0}(x) \cdot (\theta - \theta_0)$$

becomes uniformly accurate

during training.

Ok so this is the

key scaling mechanism

$$f_\theta(x) := \alpha \tilde{f}_\theta(x)$$

Define $\delta\theta := \theta - \theta_0$

$$\tilde{f}_\theta(x) = \tilde{f}_{\theta_0}(x) + J_{\theta_0}(x)\delta\theta + \cdots$$

$$J_{\theta_0}(x) := \nabla_\theta \tilde{f}_{\theta_0}(x)^T \in \mathbb{R}^{1\times p}$$

$$f_\theta(x) = \alpha \tilde{f}_{\theta_0}(x) + \alpha J_{\theta_0}(x)\delta\theta$$

Evaluating

$$\vec{f}(\theta) = \alpha \vec{\tilde{f}}_0 + \alpha J \delta\theta + \cdots$$

$$\mathcal{L}(\theta) = \frac{1}{2}\| \alpha \vec{\tilde{f}}_0 + \alpha J \delta\theta + \cdots - y \|_2^2$$

near a critical point

$$J\delta\theta = \vec{y} - \alpha\vec{\tilde{f}}_0 + \dots$$

$$\rightarrow \quad \delta\theta = \frac{1}{\alpha} J^{+}(\vec{y} - \alpha\vec{\tilde{f}}_0)$$

Assume small initialization!

$$\|\delta\theta\| = O(1/\alpha)$$

And NTK does not change

$$\Theta_0(x, x') = \alpha^2 \langle \nabla_\theta \tilde{f}_0(x), \nabla_\theta \tilde{f}_0(x') \rangle$$

If we assume (uniformly bounded Hessian)

$$\|\nabla_\theta f_0(x) - \nabla_\theta \tilde{f}_{\theta_0}(x)\| \leq H\|\theta - \theta_0\|$$

Using $\|\theta - \theta_0\| = O(1/\alpha)$

$\|\nabla_\theta \tilde{f}_\theta(x) - \nabla_{\theta_0} \tilde{f}_{\theta_0}(x)\| = O(\alpha)$

Bring it here

$$\Theta_\theta - \Theta_{\theta_0} = \alpha^2 \big(\langle \nabla \tilde{f}_\theta(x), \nabla_\theta f(x') \rangle - \langle \nabla_{\theta_0} f(x), \nabla_{\theta_0} f(x') \rangle$$

$$= \alpha^2 \big(\langle \tilde{f}_\theta(x) - \nabla \tilde{f}_{\theta_0}(x), \nabla \tilde{f}_\theta(x') \rangle + \langle \tilde{f}_{\theta_0}(x), \nabla \tilde{f}_\theta(x') - \nabla \tilde{f}_{\theta_0}(x') \rangle$$

$$\|\Theta_\theta - \Theta_{\theta_0}\| = \alpha^2 O\left(\frac{1}{\alpha}\right) = O(\alpha)$$

$$\frac{\|\theta_\theta - \theta_{\theta_0}\|}{\|\theta_{\theta_0}\|} = O(1/\alpha)$$

nearly constant

Lazy training is a scaling regime, not a structural property of NN's

Large α means:

- tiny movement in parameters
- produces larger changes in f

Informal calculus

$$\tilde{f}_{\theta_0+\delta\theta}(x) = \tilde{f}_{\theta_0}(x) + \nabla_\theta \tilde{f}_{\theta_0}(x) \cdot \delta\theta + \dots$$

multiply through by α

$$\delta f = \alpha \nabla_\theta \tilde{f}_{\theta_0} \cdot \delta\theta + \dots$$

$$|\delta f| \sim \alpha \|\delta\theta\|$$