More on NTK

Dmitry Yarotsky

Lazy training (see Chizat-Bach [1])

Assume any predictive model $\widehat{f}(\mathbf{W}, \mathbf{x})$ (e.g. a deep ANN)

Training by GD with quadratic loss:

$$\frac{d}{dt}\mathbf{W}(t) = -\nabla_{\mathbf{W}} L = -\int_X (\widehat{f}(\mathbf{W}(t), \mathbf{x}) - f(\mathbf{x}))\nabla_{\mathbf{W}}\widehat{f}(\mathbf{W}(t), \mathbf{x})d\mu(\mathbf{x})$$

Key assumption: $\mathbf{W}(t)$ remains sufficiently close to $\mathbf{W}(0)$ so that linearization is valid ("lazy training")

$$\widehat{f}(\mathbf{W}(t), \mathbf{x}) \approx \widehat{f}(\mathbf{W}(0), \mathbf{x}) + (\mathbf{W}(t) - \mathbf{W}(0)) \cdot \nabla_{\mathbf{W}}\widehat{f}(\mathbf{W}(0), \mathbf{x})$$
$$\nabla_{\mathbf{W}}\widehat{f}(\mathbf{W}(t), \mathbf{x}) \approx \nabla_{\mathbf{W}}\widehat{f}(\mathbf{W}(0), \mathbf{x})$$

Yields a well-understood linear evolution equation (**exercise**):

$$\frac{d}{dt}\mathbf{W}(t) = -A\mathbf{W}(t) + \mathbf{b}, \quad A \geq 0$$

[1] L. Chizat and F. Bach, A Note on Lazy Training in Supervised Differentiable Programming, arXiv:1812.07956

When does lazy training occur?

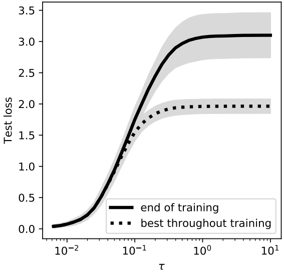
Rescale the predictive model and loss function:

$$L_\alpha(\mathbf{W}) := \frac{1}{\alpha^2} L(\alpha \widehat{f}(\mathbf{W}, \cdot)) = \frac{1}{2} \int_X (\widehat{f}(\mathbf{W}, \mathbf{x}) - \alpha^{-1} f(\mathbf{x}))^2 d\mu(\mathbf{x})$$

At large α, $\alpha^{-1} f(\mathbf{x}) \approx 0$, so if $\widehat{f}(\mathbf{W}(t=0), \cdot) = 0$, then $\mathbf{W}(t) \approx \mathbf{W}(0)$ for all t – lazy training!

Lazy training does not exploit nonlinearities and typically is less efficient than full training

(here, $\tau \sim \alpha$)



NTK and the Hessian of the loss

NTK:

$$\Theta = JJ^T, \quad J_{ij} = \frac{\partial \widehat{f}(\mathbf{W}, \mathbf{x}_i)}{\partial w_j}$$

Consider the quadratic loss: $L(\mathbf{W}) = \frac{1}{2}\sum_{k=1}^{N}(\widehat{f}(\mathbf{W}, x_k) - y_k)^2$
Then the Hessian

$$H_{ij} = \frac{\partial^2 L(\mathbf{W})}{\partial w_i \partial w_j} = \sum_{k=1}^{N} \frac{\partial \widehat{f}(\mathbf{W}, \mathbf{x}_k)}{\partial w_i} \frac{\partial \widehat{f}(\mathbf{W}, \mathbf{x}_k)}{\partial w_j} + \sum_{k=1}^{N} (\widehat{f}(\mathbf{W}, x_k) - y_k) \frac{\partial^2 \widehat{f}(\mathbf{W}, \mathbf{x}_k)}{\partial w_i \partial w_j}$$

$$\approx (J^T J)_{ij} \qquad \text{if } \widehat{f}(\mathbf{W}, x_k) \approx y_k \text{ for all } k$$

Exercise: Matrices JJ^T and $J^T J$ have the same eigenvalues, with possible exception for the eigenvalue 0.

Thus, near the global minimum $\mathbf{w}_*$ with $L(\mathbf{w}_*) \approx 0$, H has approximately the same spectrum as the NTK except maybe for eigenvalue 0.

Beyond the NTK regime: the catapult mechanism[2]

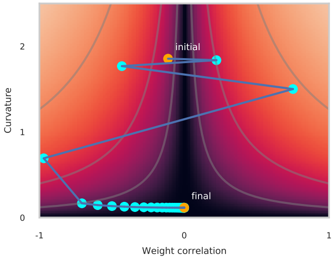
GD with learning rate η : $\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \nabla_{\mathbf{W}} L$

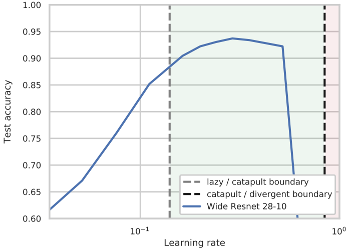
Exercise: Let $L(\mathbf{W}) = \frac{1}{2}(\mathbf{W} - \mathbf{W}_0)^T Q(\mathbf{W} - \mathbf{W}_0)$, $Q \succeq 0$. Then GD converges iff $\eta < \eta_{\text{crit}} = 2/\lambda_0$, where λ_0 is the largest eigenvalue of Q.

Three phases for more general L:

1. $\eta < \eta_{\text{crit}}$: lazy phase
2. $\eta_{\text{crit}} < \eta < \eta_{\max}$: "catapult" phase
3. $\eta_{\max} < \eta$: divergent phase

$\eta_{\max} = c/\lambda_0$, where $c \approx 4 - 12$

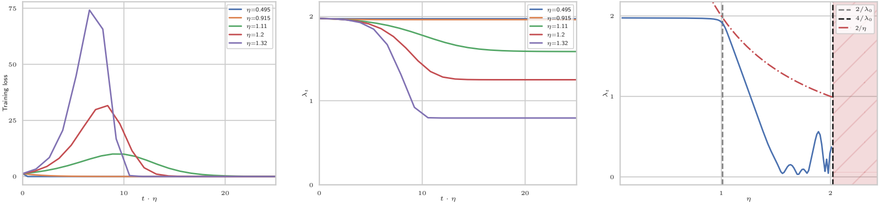




[2] Lewkowycz et al., arXiv:2003.02218

The catapult phase

- GD starts by diverging (since $\eta > \eta_{\text{crit}}$), and leaves the lazy regime; the NTK starts to change
- The largest eigenvalue $\lambda_0(t)$ of the NTK decreases, so that $\eta < \eta_{\text{crit}}(t)$ at sufficiently large t
- GD enters another lazy regime and converges to a solution with low $\lambda_0(t)$ (i.e. low loss curvature and presumably good generalization)



A toy model

A linear two-layer network with width n, approximating univariate $y(x)$:

$$f = n^{-1/2}\mathbf{v}^T\mathbf{u}x, \quad L(\mathbf{u},\mathbf{v}) = (f-y)^2/2$$

Consider single-point training set $(x=1, y)$, let $\Delta f = f - y$

Exercise:

1. The GD iterations are

$$\mathbf{u}_{t+1} = \mathbf{u}_t - \eta n^{-1/2}\Delta f_t \mathbf{v}_t, \quad \mathbf{v}_{t+1} = \mathbf{v}_t - \eta n^{-1/2}\Delta f_t \mathbf{u}_t$$

2. The NTK is $\lambda = n^{-1}(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2)$
3. For finite n, evolution of Δf_t and λ_t is given **exactly** by

$$\Delta f_{t+1} = (1 - \eta\lambda_t + \eta^2\Delta f_t^2/n)\Delta f_{t+1}, \quad \lambda_{t+1} = \lambda_t + \eta(\eta\lambda_t - 4)\Delta f_t^2/n$$

4. Let $2/\lambda_0 < \eta < 4/\lambda_0$. Then λ_t monotonically decreases; $|\Delta f_t|$ first increases, then decreases to 0.

Generalization performance with realistic models/datasets