The Large Learning Rate Phase of Deep Learning

Yasaman Bahri
Google

Aitor Lewkowycz

Ethan Dyer

Jascha Sohl-Dickstein

Guy Gur-Ari

Broad goals in science of deep learning

Understand how deep neural networks learn

- How does algorithm, architecture, hyperparameters, choice of task play a role in the final result?

But there's much to understand, which makes this a tricky problem. How to guide problem selection?

- Usually have something in mind: performance or generalization, uncertainty, robustness, privacy, fairness, etc.

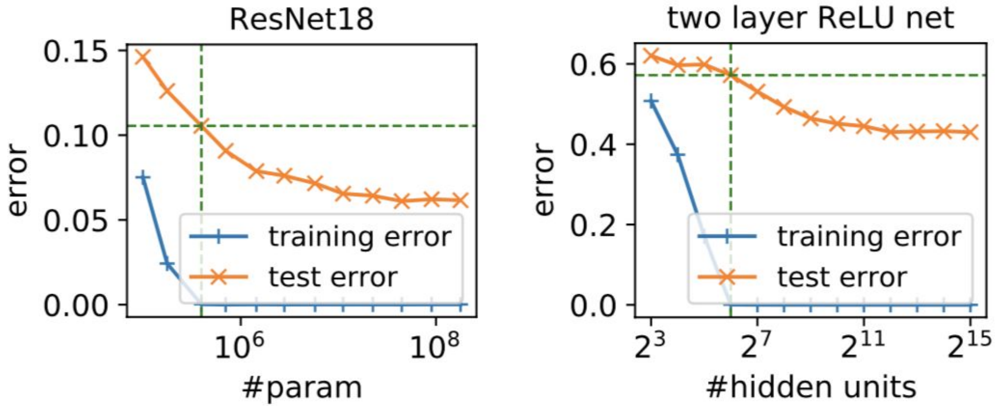
This talk:

- Motivated with generalization in mind
- Motivated by trying to partition space of hyperparameters into distinct classes
- Motivated by going beyond our previous work on the infinite width limit

Towards the limit of infinite width

Trend in deep learning has been towards overparameterization (width, depth)

Natural to ask: what happens to neural networks in the infinite width limit?



See e.g. B. Neyshabur, et al. ICLR 2015 workshop, NeurIPS 2017, ICLR 2019.

The Infinite Width Story: Gaussian Processes and Kernels

$$\text{Computation: } f_i^l(x) = b_i^l + \sum_{j=1}^{n} W_{ij}^l \phi(f_j^{l-1}(x))$$

In the infinite width limit: $f_i^l \sim \mathcal{GP}(0, K^l)$

"NNGP" Kernel

$$K^l(x, x') = \sigma_b^2 + \sigma_w^2 \mathcal{C}_\phi\left(K^{l-1}(x, x'), K^{l-1}(x, x), K^{l-1}(x', x')\right)$$

Enables exact Bayesian inference.

[1]. R. Neal. "Priors for Infinite Networks." 1994. [Single-hidden layer neural network]
[2]. Lee* and **YB***, et al. ICLR 2018. [Deep neural networks]
[3]. A. G. de G. Matthews, et al. ICLR 2018. [Deep neural networks]
Architecture dependent extensions by many others not listed, including conv, attention, graph NNs.
Recently, G. Yang, NeurIPS 2019. [General architectures]
[4]. S. Yaida. PMLR 2020. [Corrections to GP prior, Bayesian inference]

The Infinite Width Story: Gradient Descent

Parameters $\{\theta_\mu\}$, scalar function $f(x)$, loss $\mathcal{L}$, training inputs x_α in set $\mathcal{D}$

Given some evolution of neural network parameters, how does the (end-to-end) function evolve? $\frac{d\theta_\mu}{dt} \to \frac{df(x)}{dt}$

$$\frac{d\theta_\mu}{dt} = -\eta \frac{\partial \mathcal{L}}{\partial \theta_\mu} = -\eta \sum_{\alpha \in \mathcal{D}} \frac{\partial \mathcal{L}}{\partial f(x_\alpha)} \frac{\partial f(x_\alpha)}{\partial \theta_\mu}$$

$$\frac{df(x)}{dt} = \sum_\mu \frac{\partial f(x)}{\partial \theta_\mu} \frac{\partial \theta_\mu}{\partial t} = -\eta \sum_{\alpha \in \mathcal{D}} \frac{\partial \mathcal{L}}{\partial f(x_\alpha)} \left(\sum_\mu \frac{\partial f(x_\alpha)}{\partial \theta_\mu} \frac{\partial f(x)}{\partial \theta_\mu} \right)$$

This equation is not closed in general.

$$\boxed{\frac{df(x)}{dt} = -\eta \sum_{\alpha \in \mathcal{D}} \frac{\partial \mathcal{L}}{\partial f(x_\alpha)} \Theta_t(x_\alpha, x)}$$

The Infinite Width Story: Gradient Descent

This highlights a special dynamical variable:

$$\Theta_t(x, x') \equiv \sum_\mu \frac{\partial f(x)}{\partial \theta_\mu} \frac{\partial f(x')}{\partial \theta_\mu}$$

It turns out that in the limit of infinite width*, this dynamical variable does not evolve -- it is frozen at its initial value ("Neural Tangent Kernel"). [1]

Gradient descent in such infinitely wide deep nets → (fixed) kernel regression.

*Under certain conditions.

[1]. See A. Jacot, et al. "Neural Tangent Kernel." NeurIPS 2018, and many others not listed here.

The View from Infinite Width

In parameter space, is equivalent to training a first-order Taylor expansion (I'll refer to as "linearization") of the model about its initial parameters.

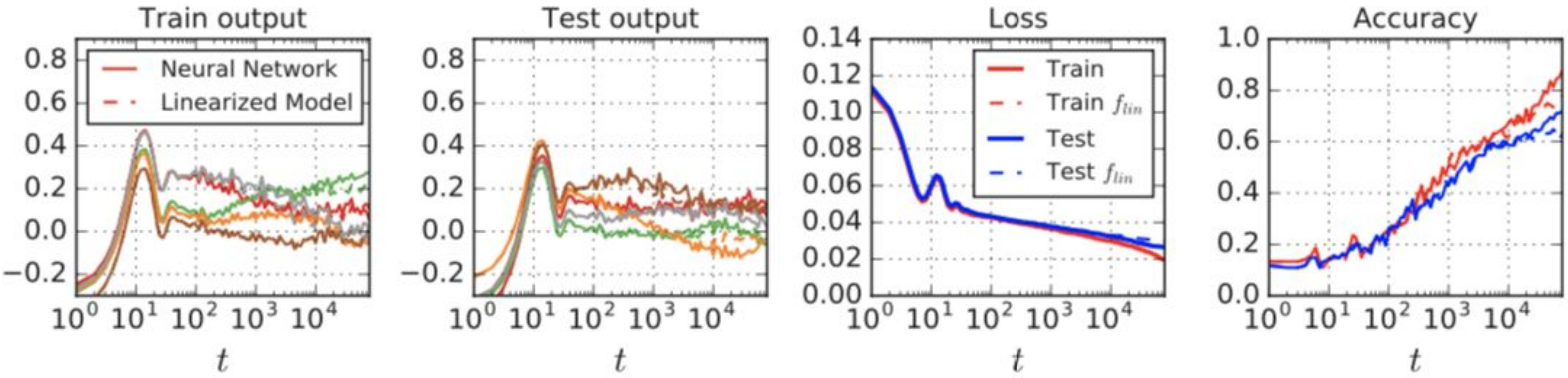
$$f_t(x) = f_0(x) + \nabla_\theta f_0(x)^T(\theta_t - \theta_0)$$

(Highlights an example of correspondence between kernels ↔ linear models constructed from their features)

[1]. Lee*, Xiao*, Schoenholz, **YB**, Novak, Sohl-Dickstein, Pennington. NeurIPS 2019.
[2]. Chizat, Oyallon, Bach. NeurIPS 2019.

Wide networks and their linearization

Which nonlinear models are well described by their linearization?



A WideResnet type model and its linearization. SGD with momentum and MSE loss on **full CIFAR-10**. Channel size = 1024, one block, batch size = 8.

This Talk

(Specializing to the case of MSE loss for remainder)

- Nonlinear models often perform better than their linearized counterparts.

- **We observed empirically:** At finite width, nonlinear models are trainable up to larger learning rates than are inaccessible for the linearized model. In many practical settings, we often tend to use large learning rates.

  - The infeasibility of the linearized problem ~ convex optimization.
  - Can we say more about the infeasibility of the nonlinear problem?
  - What happens to the nonlinear model in this other learning rate regime, since it cannot behave as a linearized model?

Partition the space of (Models + SGD)

If you trained the same model at different learning rates, what would you observe?

learning rate η

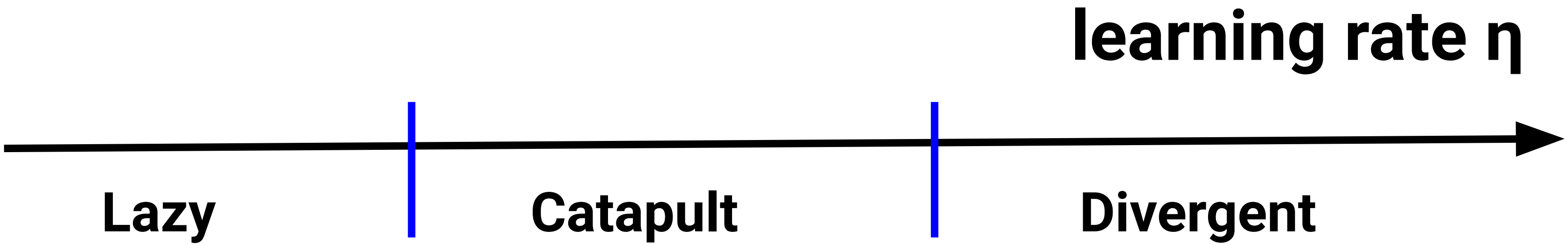
"Small"? **"Large"?** **"Divergent"?**

Special quantity λ_0

(This is the top eigenvalue of the NTK *at initialization*, which you can think of as ≈ the top eigenvalue of Hessian. The two are exactly the same at infinite width, specializing to MSE loss.)

$$H_{\mu\nu} = \frac{\partial^2 \mathcal{L}}{\partial\theta_\mu \partial\theta_\nu} = \sum_\alpha \frac{\partial f(x_\alpha)}{\partial\theta_\mu}\frac{\partial f(x_\alpha)}{\partial\theta_\nu} + \sum_\alpha (f(x_\alpha) - y_\alpha)\frac{\partial^2 f(x_\alpha)}{\partial\theta_\mu \partial\theta_\nu}$$

Delineation of Phases



$\eta_{crit} = 2/\lambda_0$

$\eta_{max} = c/\lambda_0$ where $c \sim 4, 12$ (mildly architecture dependent)

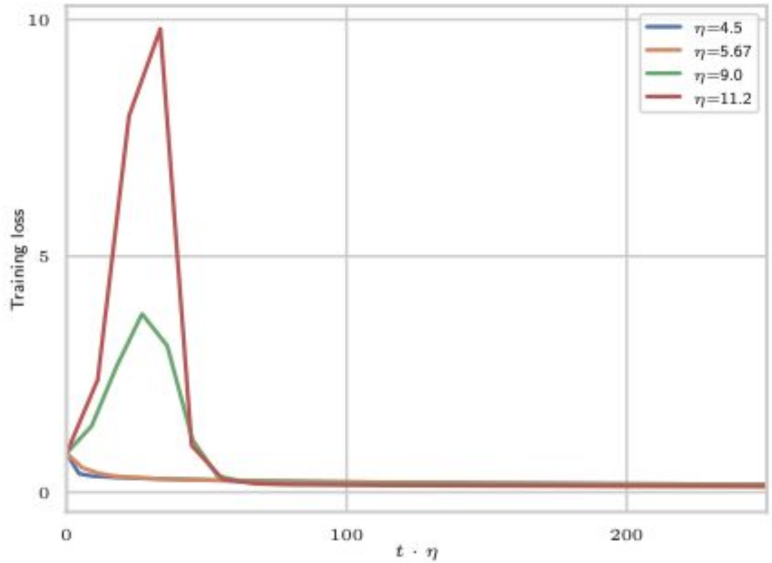
Connects to existing infinite-width theory as networks become wider

Appears to not connect to existing infinite-width theory

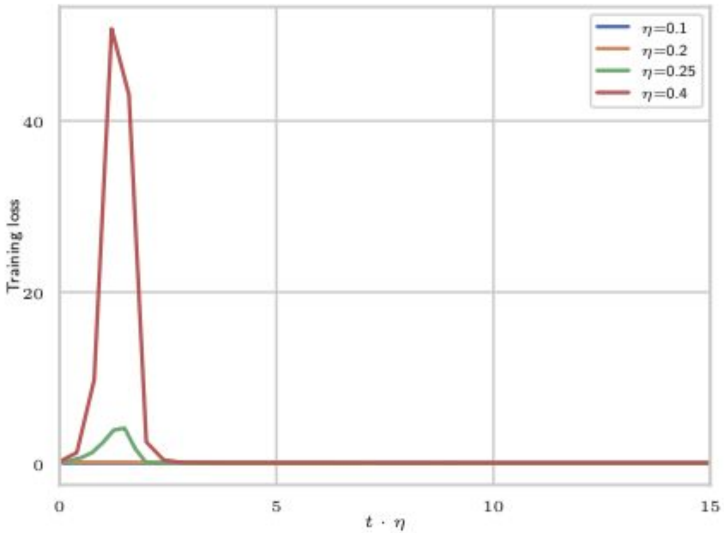
Signature: evolution of the loss (train, test)

$\eta_{crit} \sim 6.25 = 2/\lambda_0$

$\eta_{crit} \sim 0.18 = 2/\lambda_0$



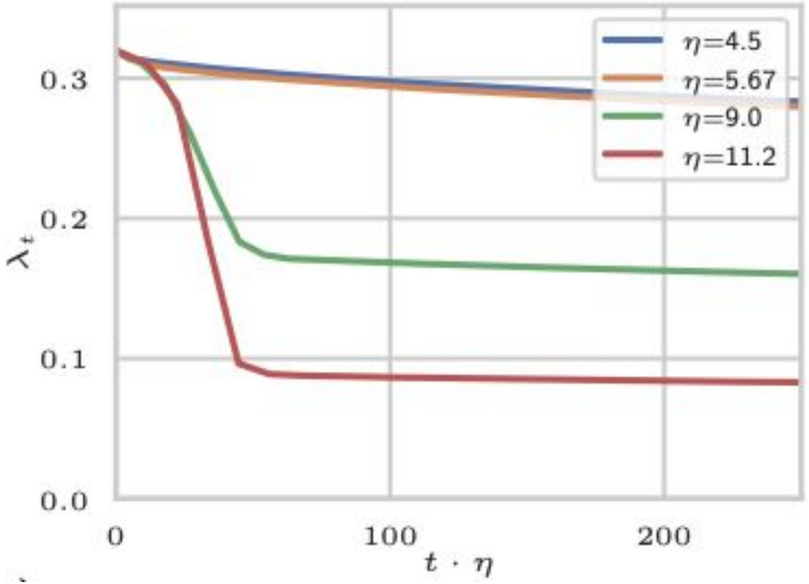
Left: Three hidden-layer Relu fully-connected network on MNIST



Right: Wide Resnet 28-10 on CIFAR-10

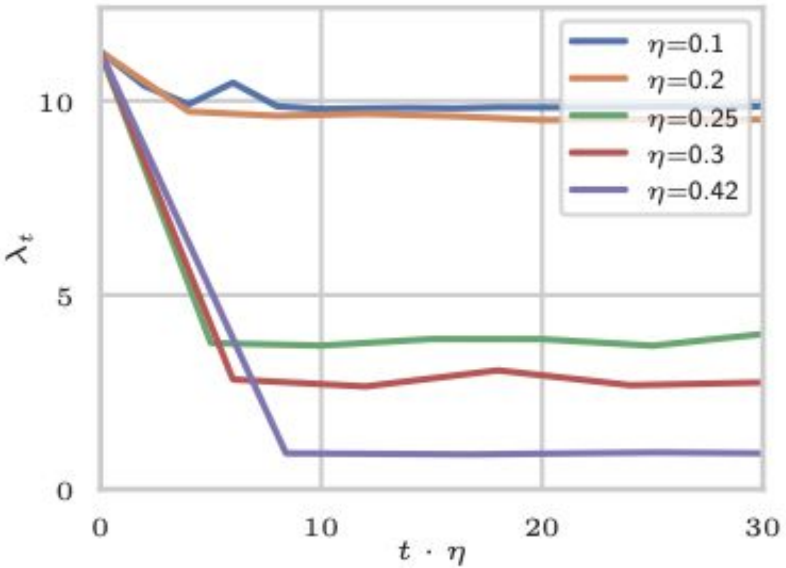
Signature: evolution of the curvature

$\eta_{crit} \sim 6.25 = 2/\lambda_0$



Left: Three hidden-layer Relu fully-connected network on MNIST

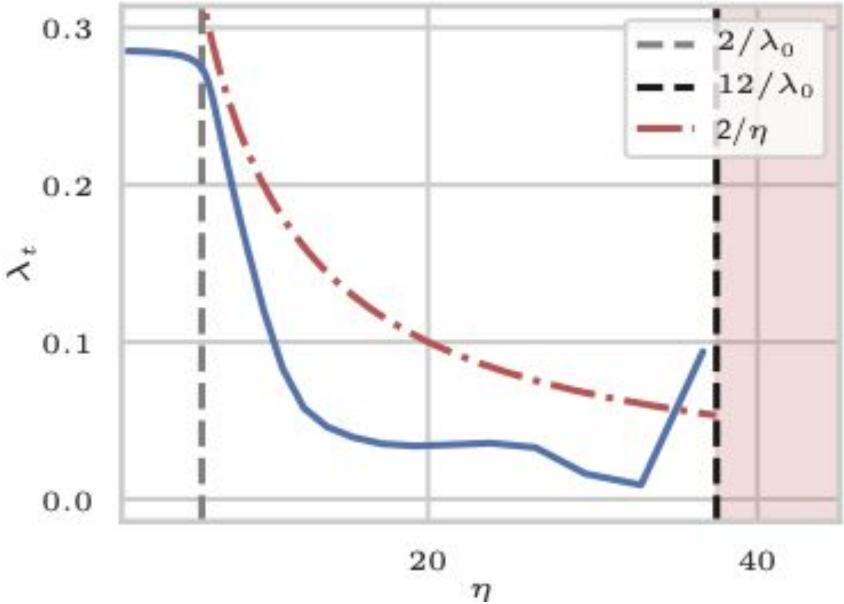
$\eta_{crit} \sim 0.18 = 2/\lambda_0$



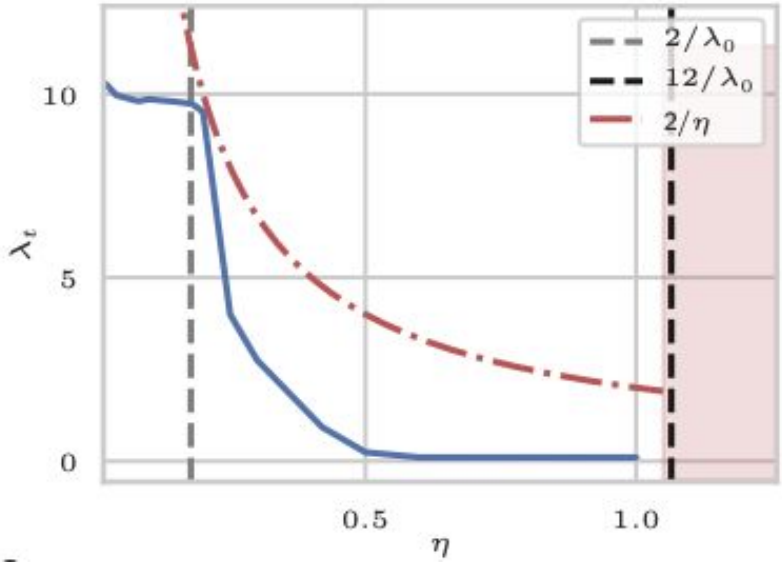
Right: Wide Resnet 28-10 on CIFAR-10

Signature: final curvature vs initial learning rate

$\eta_{crit} \sim 6.25 = 2/\lambda_0$



$\eta_{crit} \sim 0.18 = 2/\lambda_0$



Left: Three hidden-layer Relu fully-connected network on MNIST

Right: Wide Resnet 28-10 on CIFAR-10

Three learning rate regimes

Lazy Phase: $\eta < 2/\lambda_0$
The curvature remains ~constant during the initial part of training. Model behaves (loosely) as a model linearized about its initial parameters (exactly true in the infinite width limit).

Catapult Phase: $\eta_{crit} = 2/\lambda_0 < \eta < \eta_{max}$
The curvature at initialization is too high for training converge to a nearby point. The linearized approximation breaks down. Training begins with a period of growth in the loss + simultaneous decrease in the curvature until it stabilizes with $\lambda_t < 2/\eta$. Converge to a flatter minimum.

We find $\boldsymbol{\eta_{max} \sim c/\lambda_0}$ where c is an architecture-dependent constant. c = 4 in the simple model, c~ 4 for Tanh networks empirically, c ~ 12 for Relu networks empirically.

Divergent Phase: $\eta > \eta_{max}$
Training diverges.

Aside: two ways to parameterize your neural network

"NTK" parameterization: Initialize $W_{ij} \sim \mathcal{N}(0, \sigma_w^2)$ and parameterize model as

$$f_i^l(x) = \sum_{j=1}^{n} \frac{1}{\sqrt{n}} W_{ij} f_j^{l-1}(x)$$

That is, explicitly factor out (width) dimensions.

"Standard" parameterization: Initialize $W_{ij} \sim \mathcal{N}(0, \sigma_w^2/n)$ and parameterize model as

$$f_i^l(x) = \sum_{j=1}^{n} W_{ij} f_j^{l-1}(x)$$

That is, have dimensions absorbed into the parameters.

(For some discussion on this, see e.g. Park, et al. arxiv 1905.03776.)

Dynamics in a simple model

Let the model be $f : \mathbb{R}^d \to \mathbb{R}$, parameters $\theta \in \mathbb{R}^p$, training set $\{(x_\alpha, y_\alpha)\}_{\alpha=1}^m$, and MSE loss

$$\mathcal{L} = \frac{1}{2m} \sum_{\alpha=1}^{m} (f(x_\alpha) - y_\alpha)^2$$

Define the NTK $\Theta : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as

$$\Theta(x, x') \equiv \frac{1}{m} \sum_{\mu=1}^{p} \frac{\partial f(x)}{\partial \theta_\mu} \frac{\partial f(x')}{\partial \theta_\mu}$$

Model is a single-hidden layer network with width n, parameters $v \in \mathbb{R}^n$ and $u \in \mathbb{R}^{n \times d}$, input $x \in \mathbb{R}^d$:

$$f(x) = \frac{1}{\sqrt{n}} v^T u x \qquad \text{(NTK param)}$$

Dynamics in a simple model

Let's specialize to a single (1D) training example $(x, y) = (1, 0)$.

$$\mathcal{L} = \frac{f^2}{2} \qquad f = \frac{v^T u}{\sqrt{n}} \qquad \Theta(1,1) = \lambda = \frac{||v||_2^2 + ||u||_2^2}{n}$$

Gradient descent updates for the parameters $(u, v \in \mathbb{R}^n)$ are

$$u_{t+1} = u_t - \frac{\eta}{\sqrt{n}} f_t v_t \qquad v_{t+1} = v_t - \frac{\eta}{\sqrt{n}} f_t u_t$$

In function space, the updates are

$$f_{t+1} = \left(1 - \eta\lambda_t + \frac{\eta^2 f_t^2}{n}\right) f_t \qquad \lambda_{t+1} = \lambda_t + \frac{\eta f_t^2}{n}(\eta\lambda_t - 4)$$

These equations are closed.
Note also, at initialization $f_0, \lambda_0 \sim \mathcal{O}(n^0) = \mathcal{O}(1)$.

Phases in a simple model

$$\boxed{f_{t+1} = \left(1 - \eta\lambda_t + \frac{\eta^2 f_t^2}{n}\right) f_t}$$

$$\boxed{\lambda_{t+1} = \lambda_t + \frac{\eta f_t^2}{n}(\eta\lambda_t - 4)}$$

Define $\eta_{\text{crit}} \equiv 2/\lambda_0$. In the infinite width limit:
$f_{t+1} = (1 - \eta\lambda_t) f_t, \quad \lambda_{t+1} = \lambda_t$. Usual NTK dynamics.

At large but finite width: when $\eta < \eta_{\text{crit}}$, note that $|1 - \eta\lambda_t| < 1$.
$\Rightarrow f, \mathcal{L}$ are shrinking. λ_t doesn't change much.
Convergence happens in $\mathcal{O}(n^0) = \mathcal{O}(1)$ steps.

Phases in a simple model

$$f_{t+1} = \left(1 - \eta\lambda_t + \frac{\eta^2 f_t^2}{n}\right) f_t$$

$$\lambda_{t+1} = \lambda_t + \frac{\eta f_t^2}{n}(\eta\lambda_t - 4)$$

Catapult phase. Consider $\frac{2}{\lambda_0} < \eta < \frac{4}{\lambda_0}$.

- $(\eta\lambda_t - 4)$ term is negative. λ_t will start to decrease but updates are small.
- Because $|1 - \eta\lambda_t| > 1$, f_t will start to grow. After $t \sim \log(n)$, $|f_t| \sim \sqrt{n}$.
- λ_t receives $\mathcal{O}(1)$ updates and will continue to drop until $\lambda_t \lesssim 2/\eta$.
- When this happens, $|1 - \eta\lambda_t| < 1$, $f, \mathcal{L}$ can resume shrinking.

Divergent phase. $\eta_{max} = \frac{4}{\lambda_0}$. Explicitly we have $c = 4$ in this model.

Three phases: catapult phase

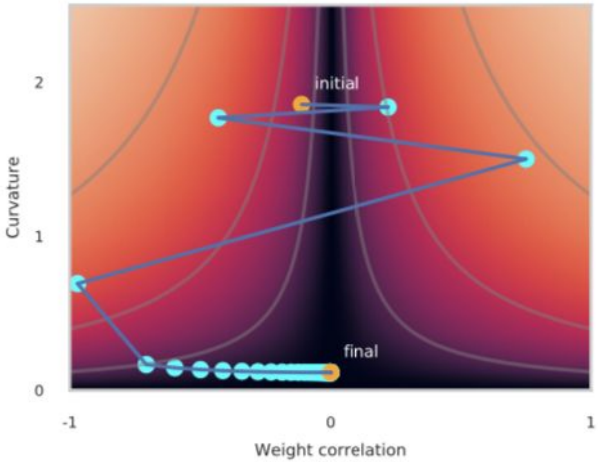
$$f_{t+1} = \left(1 - \eta\lambda_t + \frac{\eta^2 f_t^2}{n}\right) f_t$$

$$\lambda_{t+1} = \lambda_t + \frac{\eta f_t^2}{n}(\eta\lambda_t - 4)$$

If we take the infinite width limit first, we will miss a stable fixed point of the dynamics different than NTK.

Remarks:

- Access in a modified notion of large width limit.
- Lower curvature at the end of training.
- Role of finite width.



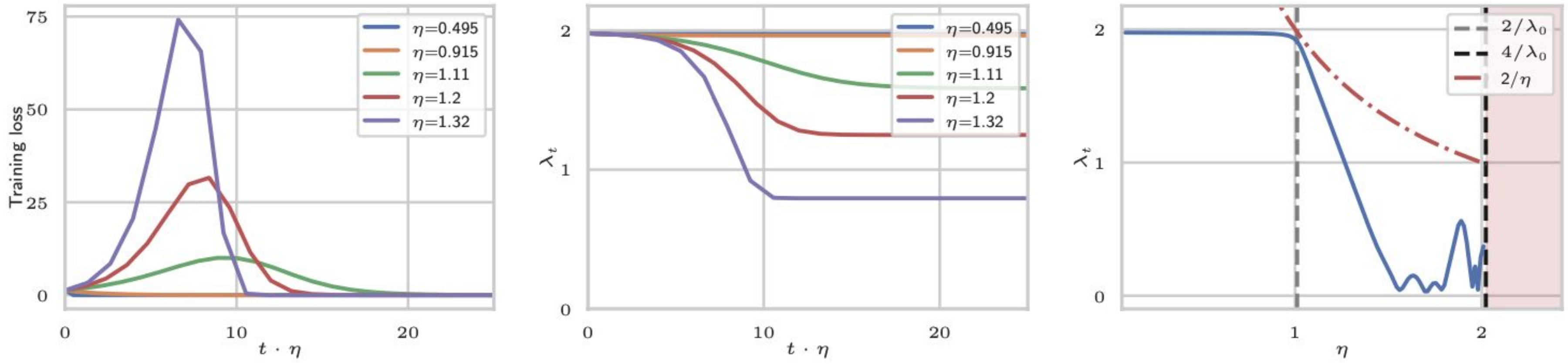
Dynamics in the catapult phase

Dynamics in a simple model

We term the period during time evolution when curvature adjusts via this mechanism the ***rearrangement***.

The numerics below are for the simple model just described.
(Here, critical η ~ 1 and width = 1000.)

We reproduce the signatures of the three phases:



Full model analysis

$$u_{ia}^{t+1} = u_{ia} - \frac{\eta}{\sqrt{nm}} v_i x_{a\alpha} \tilde{f}_\alpha \qquad v_i^{t+1} = v_i - \frac{\eta}{\sqrt{nm}} u_{ia} x_{a\alpha} \tilde{f}_\alpha$$

$$\Theta_{\alpha\beta} = \frac{1}{nm}\left(|v|^2 x_\alpha^T x_\beta + x_\alpha^T u^T u x_\beta \right)$$

Definitions: $\tilde{f}_\alpha \equiv (f(x_\alpha) - y_\alpha)$ and $\zeta \equiv \frac{1}{m} \sum_\alpha \tilde{f}_\alpha x_\alpha \in \mathbb{R}^d$

In function space, the updates are:

$$\tilde{f}_\alpha^{t+1} = (\delta_{\alpha\beta} - \eta \Theta_{\alpha\beta}) \tilde{f}_\beta + \frac{\eta^2}{nm} (x_\alpha^T \zeta)(f^T \tilde{f})$$

$$\Theta_{\alpha\beta}^{t+1} = \Theta_{\alpha\beta} - \frac{\eta}{nm} \left[(x_\beta^T \zeta) f_\alpha + (x_\alpha^T \zeta) f_\beta + \frac{2}{m} (x_\alpha^T x_\beta)(\tilde{f}^T f) \right] + \frac{\eta^2}{n^2 m} \left[|v|^2 (x_\alpha^T \zeta)(x_\beta^T \zeta) + (\zeta^T u^T u \zeta)(x_\alpha^T x_\beta) \right]$$

Full model analysis

A projected equation looks a bit more similar:

$$\tilde{f}^T \Theta_{t+1} \tilde{f} = \tilde{f}^T \Theta \tilde{f} + \frac{\eta}{n} \zeta^T \zeta \left(\eta \tilde{f}^T \Theta \tilde{f} - 4 f^T \tilde{f} \right)$$

The error vector starts to project onto the top NTK eigendirection exponentially fast, so approximate it as lying along that subspace to find:

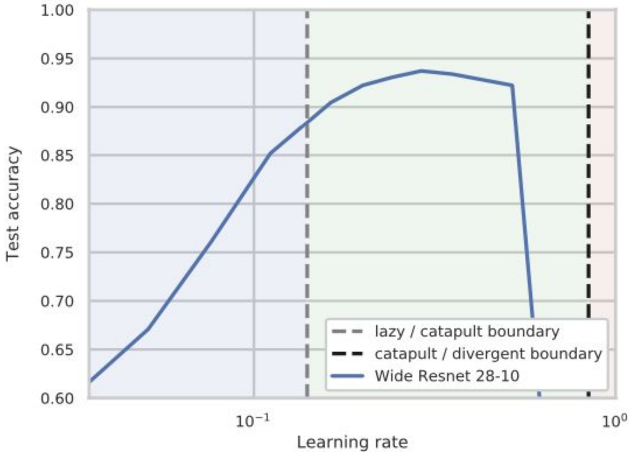
$$\lambda_{t+1} \approx \lambda + \frac{\eta}{n} \zeta^T \zeta (\eta \lambda - 4)$$

So that a similar analysis to the simplest model can be done.

Connection to generalization

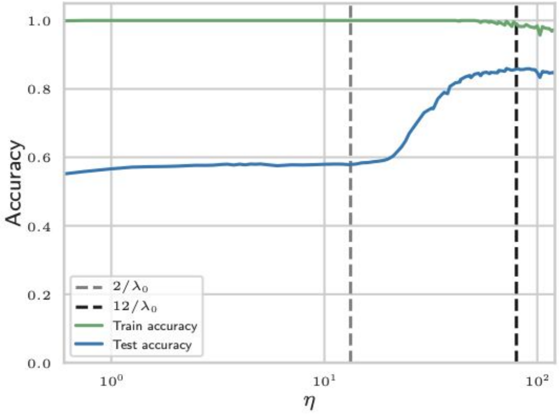
- Lazy phase and catapult phase have different behaviors in early time dynamics.
- This particularly affects the curvature.
- Empirically, we find that these differences at early times ***often*** have implications for generalization (i.e. late-time dynamics).

Comment on comparison:
- Could compare for fixed step budget.
- Could compare for same physical time budget. We find differences can still persist even when the smaller learning rates have ‘equivalent’ time.
    - Evolution for same physical time t = η * step.

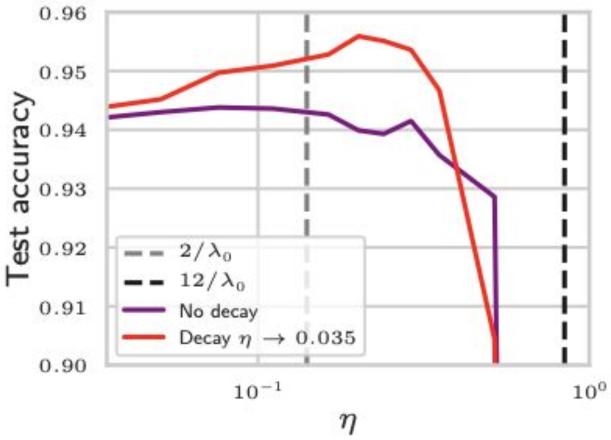


Fixed step comparison.

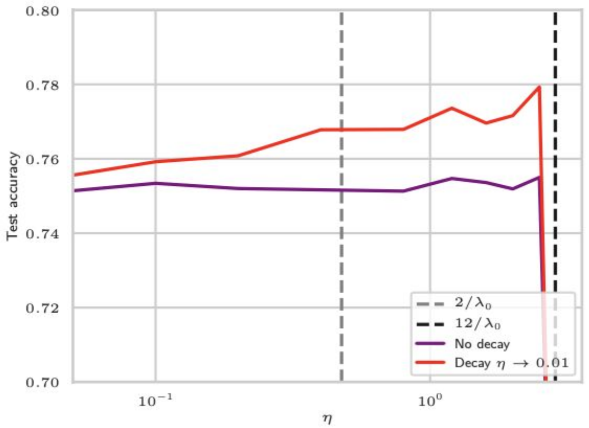
Comparison of generalization across learning rate



Single hidden-layer FC Relu on 512 MNIST samples



Wide Resnet 28-10 on CIFAR-10 with L2 reg and data augmentation



Wide Resnet 28-10 on CIFAR-100 with L2 reg and data augmentation

Larger learning rates -- lower curvature at the end of training (flatter minima) -- typically better performance

Phase transitions & perturbation theory

Schematically, we have an expansion: $f_t = f_t^{(0)} + \frac{1}{n} f_t^{(1)} + \ldots$

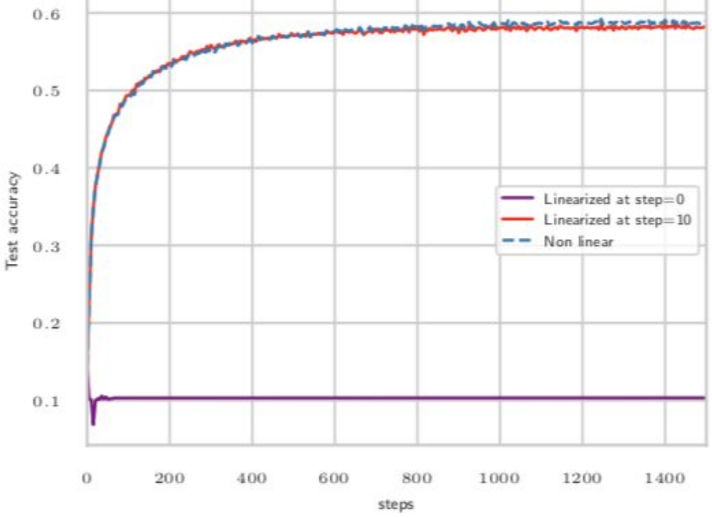
As we saw in the simple model, all terms become of ~ the same order and cannot be ignored.

Perturbation theory studied in [1]; we believe this transition is a breakdown in the expansion.

[1]. Dyer & Gur-Ari, ICLR 2020. Huang & Yau, ICML 2020.

However, once the curvature scale drops, as we saw, we can go back to ignoring those higher-order terms.

- Can resume treatment as a linearized model
- Perturbation theory with respect to a point after the rearrangement will be well-behaved



Single hidden-layer FC Relu on 512 MNIST samples, with LR in the catapult phase

Phase transition: critical exponent

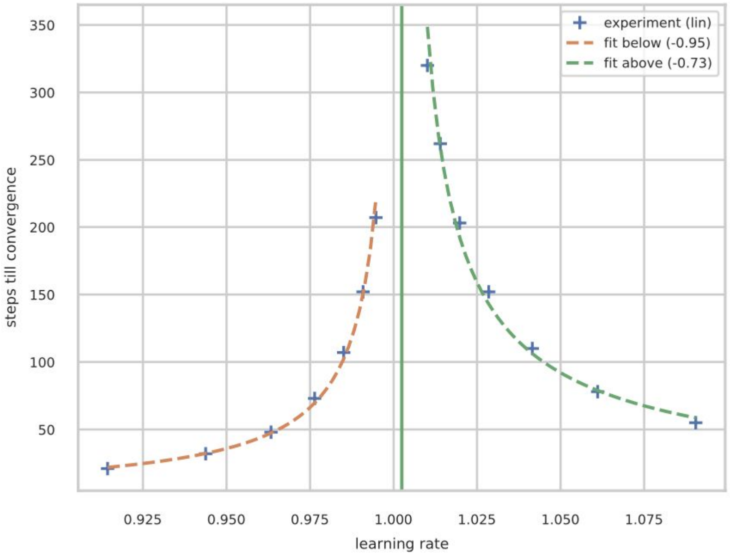
Expect non-analyticity in the final curvature as a function of learning rate (in this modified infinite width limit).

$$\lambda_*(\eta) \text{ is constant for } \eta < \eta_{crit} \text{ but decreases for } \eta > \eta_{crit}$$

Number of steps till convergence:

$$t_*(\eta) = |\eta_{crit} - \eta|^{-1}$$

Same exponent above/below the transition.



Closing Remarks

- Rather universal empirical signatures of the catapult phase across datasets, architectures
  - Growth in loss, drop in curvature, relevant time scale
  - $\eta_{crit} = 2/\lambda_0$, $\eta_{max} \sim c/\lambda_0$.
- Guide for hyperparameter tuning (when using MSE loss)
  - Only need a measurement (NTK top eigenvalue) at initialization
- Analysis of a closed dynamical system reveals different phases
  - Modified infinite-width, infinite time limit
  - Dynamical mechanism seems to be more general
- Breakdown of perturbation theory; phase transition
- Connection to generalization

arxiv 2003.02218