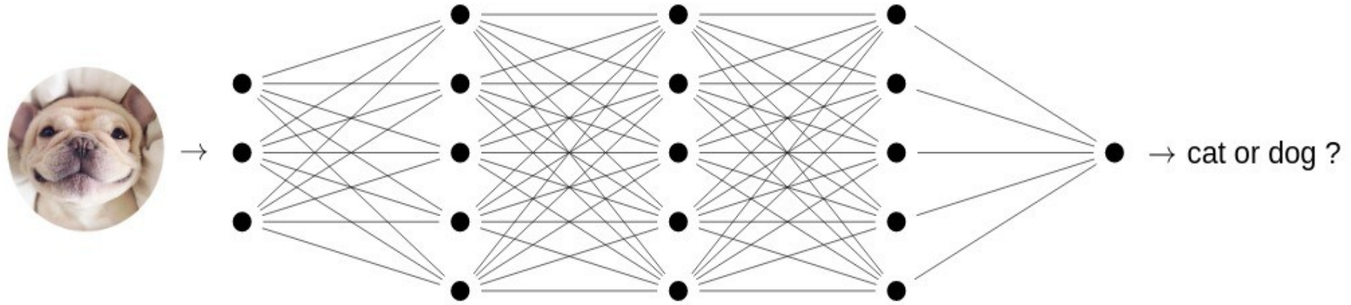
Feature vs. Lazy Training

Two different regimes in the dynamics of neural networks

Mario Geiger Stefano Spigler

Arthur Jacot Matthieu Wyart

https://arxiv.org/abs/1906.08034



$f(w, x)$

feature training	**lazy training**
the network learns features	no features are learned
	$f(w) \approx f(w_0) + \nabla f(w_0) \cdot dw$

characteristic training time separating the two regimes

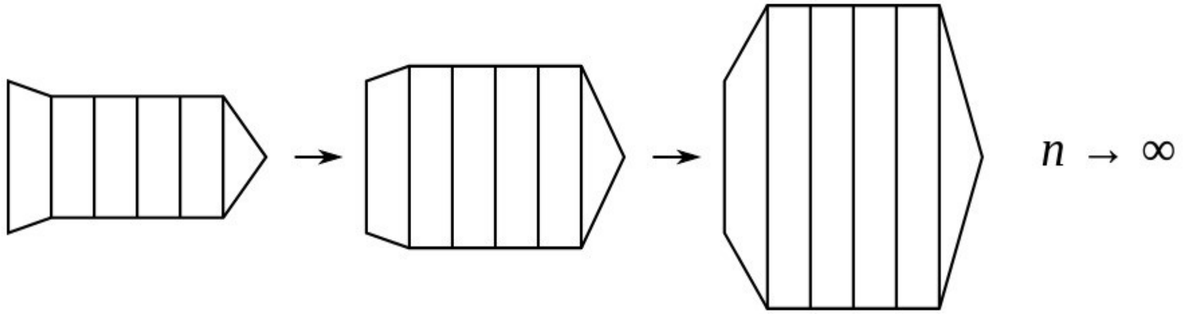
training procedure to force each regim

Overparametrization

- perform well
- theoretically tractable

How does the network perform in the infinite width limit?

There exist **two**??limits in the literature

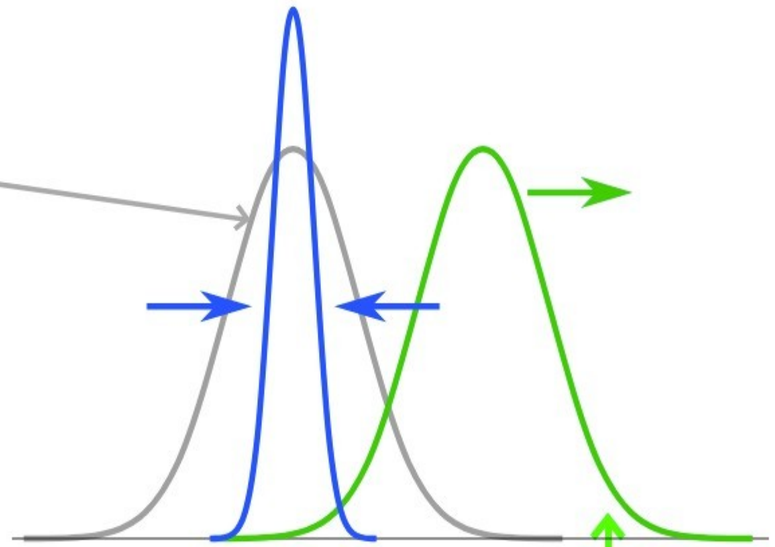


n neurons per layer

The 2 limits can be understood in the context of the central limit thm

Central Limit Theorem:

As $n \to \infty$, $Y \longrightarrow$ Gaussian



(Law of large numbers)

$$Y = \frac{1}{n} \sum_{i=1}^{n} X_i$$

$\langle Y \rangle = \langle X_i \rangle$?? ??

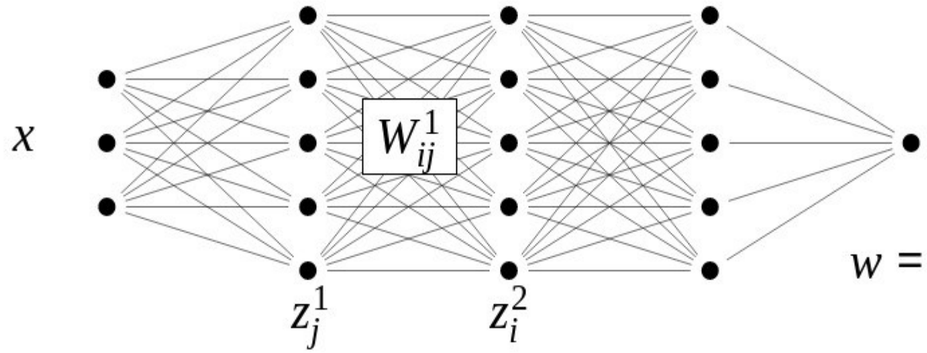
$\langle Y^2 \rangle - \langle Y \rangle^2 = \frac{1}{n}(\langle X_i^2 \rangle - \langle X_i \rangle^2)$

$$Y = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} X_i$$

$\langle Y \rangle = \sqrt{n}\langle X_i \rangle$?? ??

$\langle Y^2 \rangle - \langle Y \rangle^2 = \langle X_i^2 \rangle - \langle X_i \rangle^2$

regime 1: kernel limit

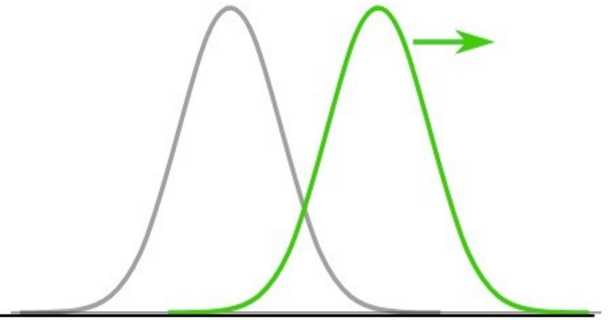


w = all the weights

At initialization

$$W_{ij} \longleftarrow \mathrm{N}(0,1)$$

$$z_i^{\ell+1} = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} W_{ij}^{\ell}\, \phi(z_j^{\ell})$$



Independent terms in the sum, CLT => when $n \to \infty$ output is a Gaussian process

[1995 R. M. Neal]

With this scaling, small correlations adds up significantly => the weights will change only a little during training

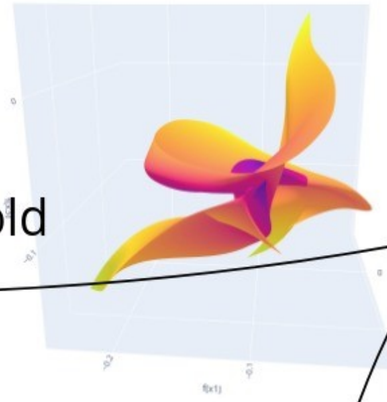
regime 1: kernel limit

space of functions $\mathrm{R}^d \to \mathrm{R}$

(dimension ∞)

network manifold

(dimension N)



$f(w)$

tangent space

(dimension N)

$f(w_0)$

$f(w_0) + \sum_\mu c_\mu \Theta(w_0, x_\mu)$

kernel space

(dimension m)

$f(w_0) + \nabla f(w_0) \cdot dw$

d input dimension

N number of parameters

m size of the trainset

regime 1: kernel limit

neural tengant kernel

$$\Theta(w, x_1, x_2) = \nabla_w f(w, x_1) \cdot \nabla_w f(w, x_2)???$$

for $n \to \infty$

The NTK is independent of the initialization
and is constant through learning
=> the network behaves like a kernel method
??

The weights barely change
$\|dw\| \sim O(1)$ and $dW_{ij} \sim O(1/n)$?? ??($O(1/\sqrt{n})$ at the extremity)
??

The internal activations barely change

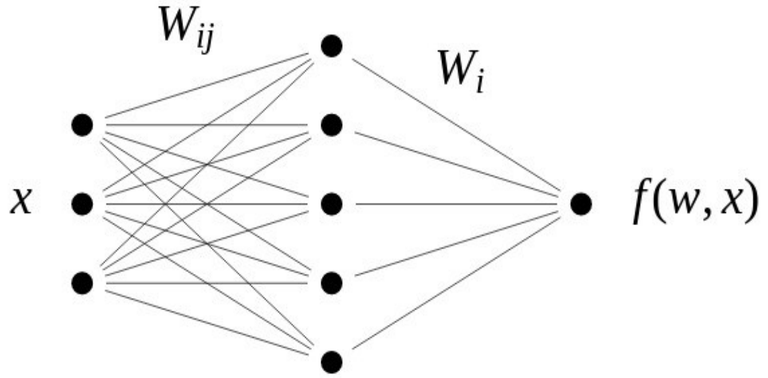
$dz \sim O(1/\sqrt{n})$
=> no feature training

[2018 Jacot et al.]
[2018 Du et al.]
[2019 Lee et al.]

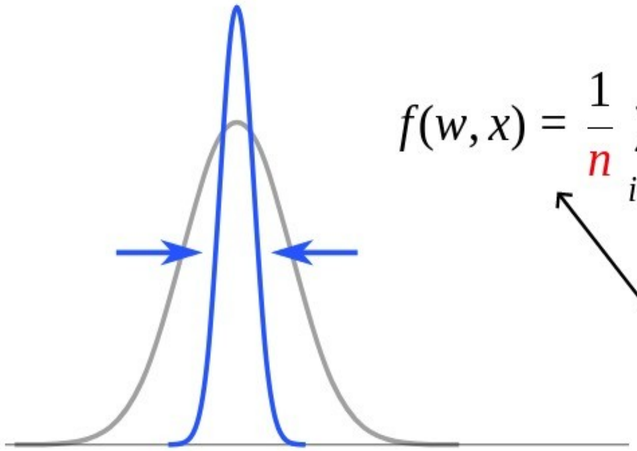
regime 2: mean field limit

Another limit !

for $n \longrightarrow \infty$



studied **theoretically** for 1 hidden layer



$$f(w,x) = \frac{1}{n} \sum_{i=1}^{n} W_i \, \phi\left(\frac{1}{\sqrt{n}} \sum_{j=1}^{n} W_{ij} x_j\right)$$

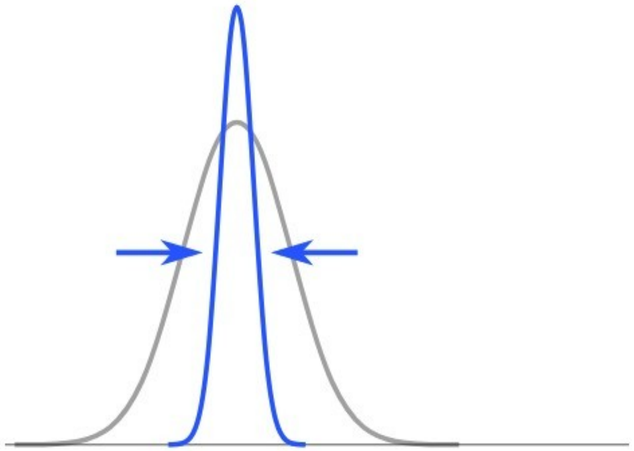
was $\frac{1}{\sqrt{n}}$ in the kernel limit

regime 2: mean field limit

$$f(w,x) = \frac{1}{n} \sum_{i=1}^{n} W_i \, \phi\left(\frac{1}{\sqrt{n}} \sum_{j=1}^{n} W_{ij} x_j\right)$$

$\frac{1}{n}$ instead of $\frac{1}{\sqrt{n}}$ implies

- no output fluctuations at initialization ??



regime 2: mean field limit

$$f(w,x) = \frac{1}{n} \sum_{i=1}^{n} W_i \, \phi\left(\frac{1}{\sqrt{n}} \sum_{j=1}^{n} W_{ij} x_j\right)$$

$\frac{1}{n}$ instead of $\frac{1}{\sqrt{n}}$ implies

- no output fluctuations at initialization ??
- we can replace the sum by an integral

$$f(w,x) \approx f(\rho,x) = \int d\rho(W,\vec{W}) \, W \, \phi\left(\frac{1}{\sqrt{n}} \sum_{j=1}^{n} W_j x_j\right)$$

where ρ is the density of neuron's weights

regime 2: mean field limit

$$f(\rho, x) = \int d\rho(W, \vec{W})\, W\, \phi\left(\frac{1}{\sqrt{n}} \sum_{j=1}^{n} W_j x_j\right)$$

ρ follows a differential equation

[2018 S. Mei et al], [2018 Rotskoff and Vanden-Eijnden], [2018 Chizat and Bach]

In this limit, the internal activation do change

$\langle Y \rangle = \langle X_i \rangle$ => feature training

What is the difference between the two limits

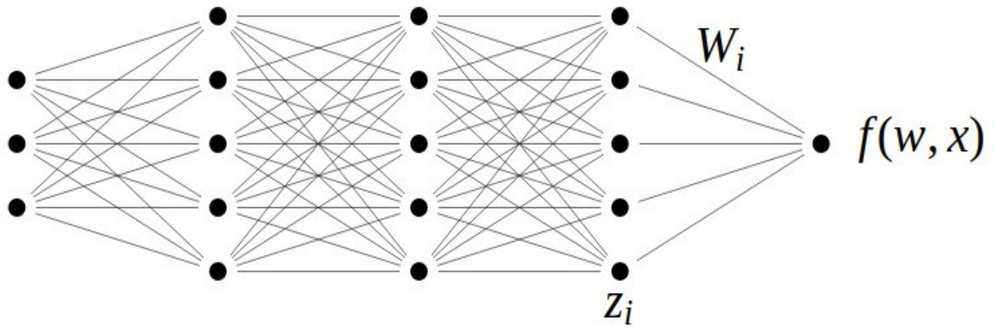
kernel limit?? ?? ?? ?? ?? ?? ?? and?? ?? ?? ?? ?? ??

$\frac{1}{\sqrt{n}}$?? **mean field limit** $\frac{1}{n}$

frozen internal activations z^{ℓ} change

- which limit describe better finite n networks?
- are there corresponding regimes for finite n???

we use the scaling factor α to investigate the transition between the two regimes



$$\alpha f(w,x) = \frac{\alpha}{\sqrt{n}} \sum_{i=1}^{n} W_i \, \phi(z_i)$$

[2019 Chizat??and Bach]

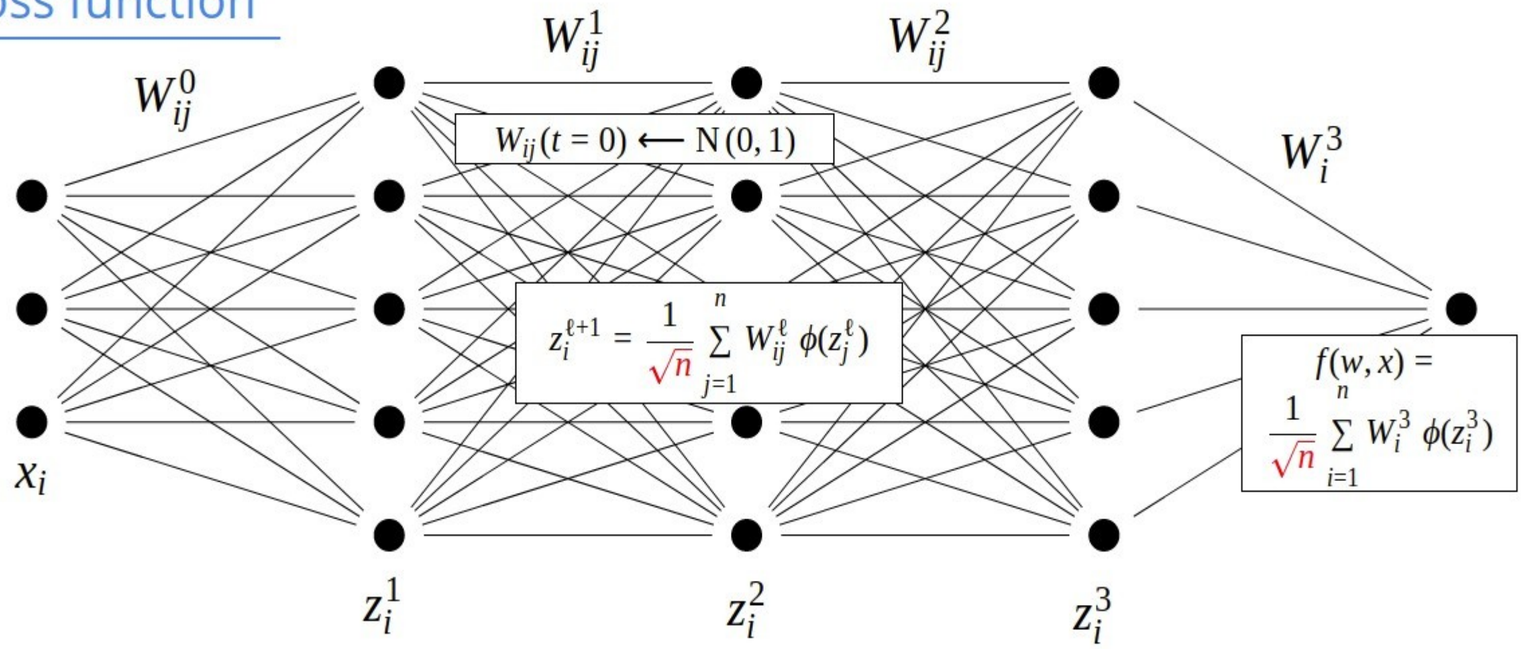
- if α is fixed constant and $n \to \infty$ then => kernel limit
- if $\alpha \sim \frac{1}{\sqrt{n}}$ and $n \to \infty$ then => mean field limit

linearize the network with $f - f_0$

We would like that for any finite n, in the limit $\alpha \to \infty$, the network behaves linearly

$$\alpha \cdot (f(w, x) - f(w_0, x))$$

loss function



$$\mathrm{L}(w) = \frac{1}{\alpha^2 |\mathrm{D}|} \sum_{(x,y)\in \mathrm{D}} \ell\left(\alpha(f(w,x) - f(w_0,x)), y\right)$$

This α^2 is here to converge in a time that does not scale with α in the limit $\alpha \to \infty$

continuous dynamics

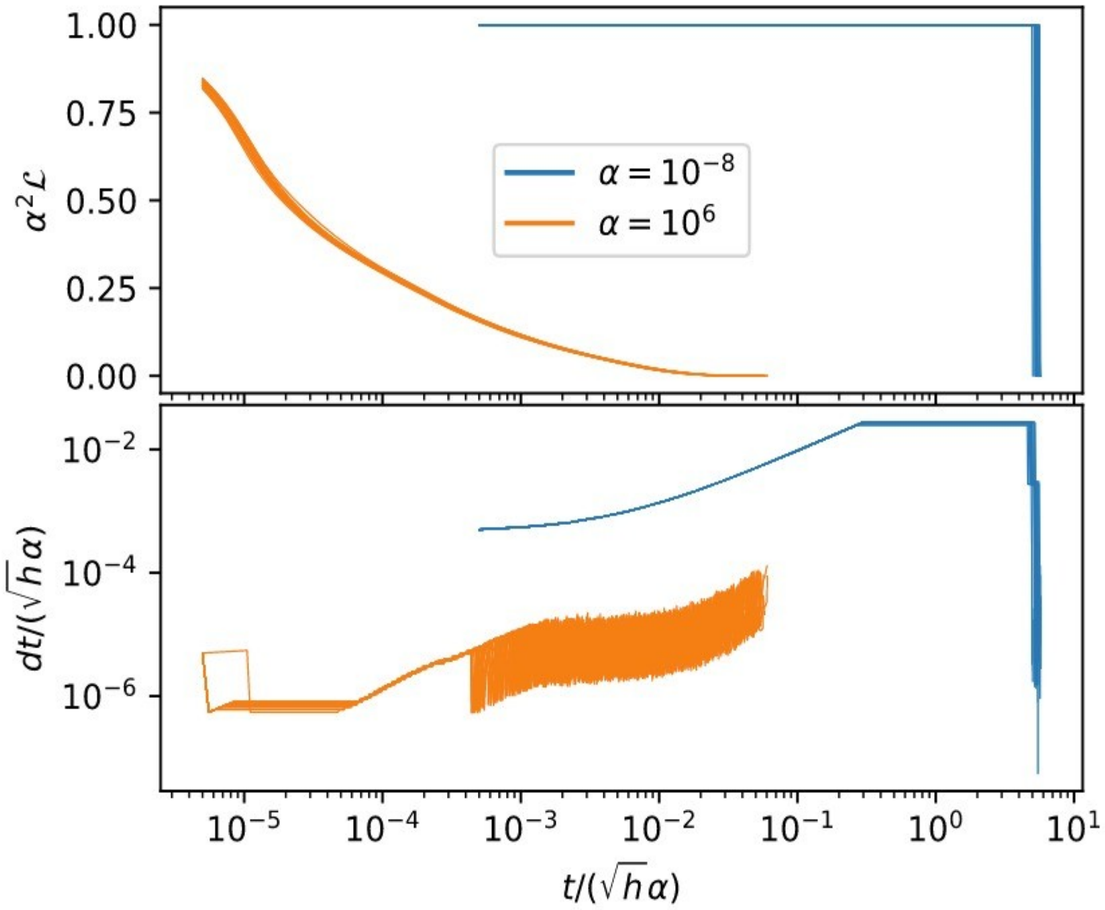
$$\dot{w} = -\nabla_w \mathrm{L}(w)$$

Implemented with a dynamical adaptation of the time step dt such that,

$$10^{-4} < \frac{\|\nabla \mathrm{L}(t_{i+1}) - \nabla \mathrm{L}(t_i)\|^2}{\|\nabla \mathrm{L}(t_{i+1})\| \cdot \|\nabla \mathrm{L}(t_i)\|} < 10^{-2}$$

(works well only with full batch and smooth loss)

continuous dynamics



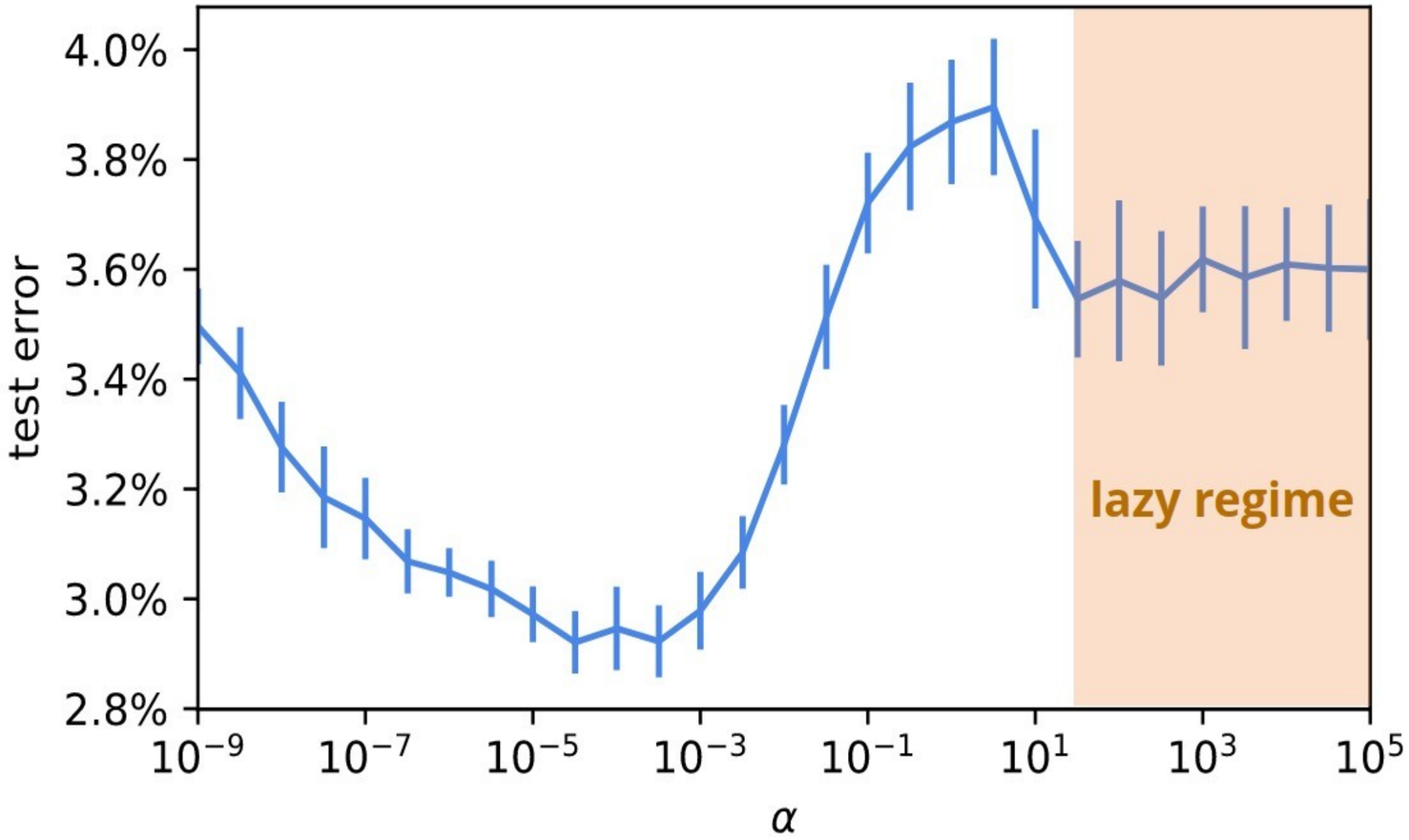
$$10^{-4} < \frac{\|\nabla \mathrm{L}(t_{i+1}) - \nabla \mathrm{L}(t_i)\|^2}{\|\nabla \mathrm{L}(t_{i+1})\| \cdot \|\nabla \mathrm{L}(t_i)\|} < 10^{-2}$$

momentum dynamics

$$\dot{v} = -\frac{1}{\tau}(v + \nabla \mathrm{L})$$

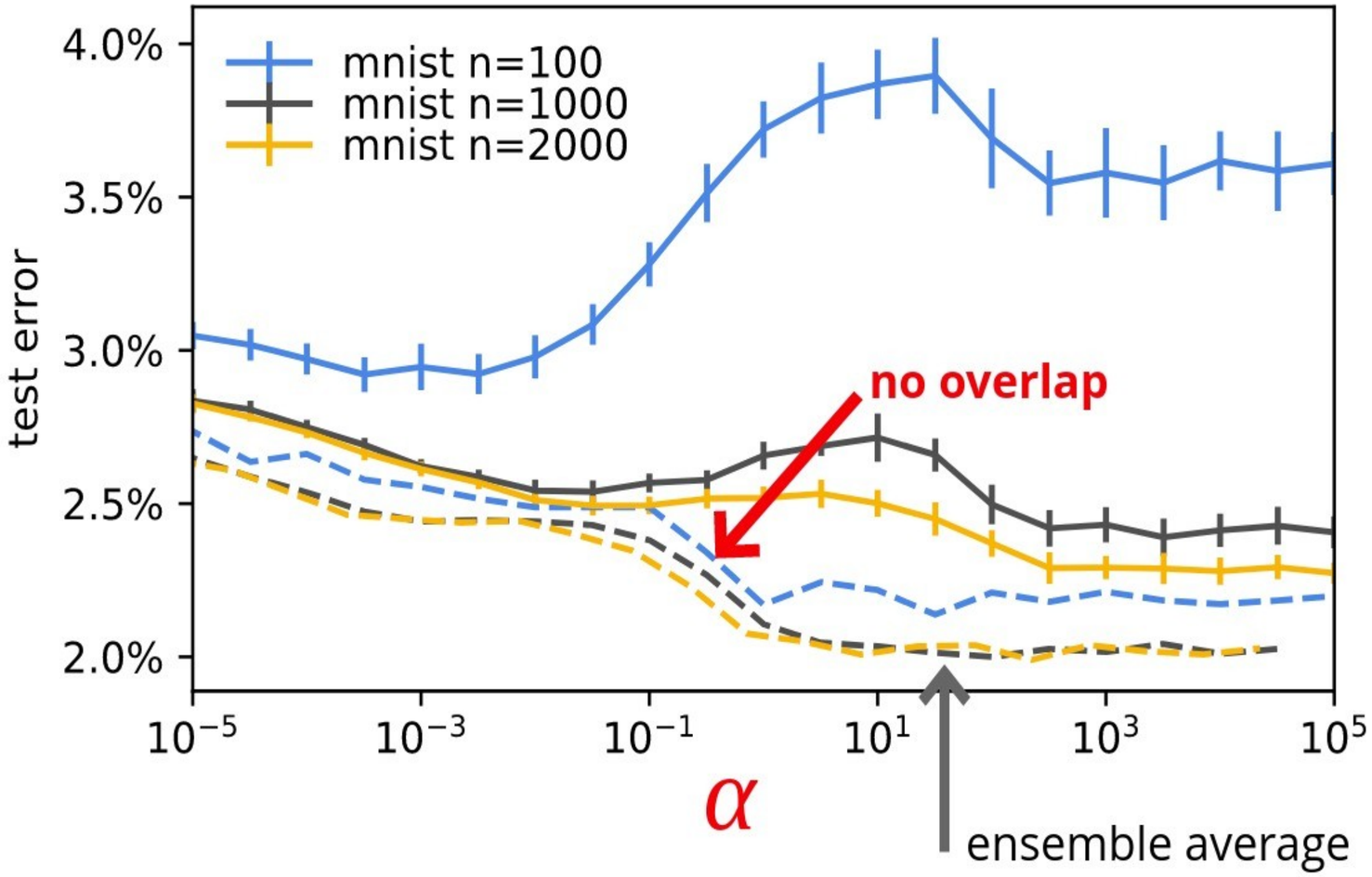
$$\dot{w} = v$$

there is a plateau for large values of α



MNIST 10k parity, FC L=3, softplus, gradient flow with momentum

the ensemble average converge with $n \to \infty$

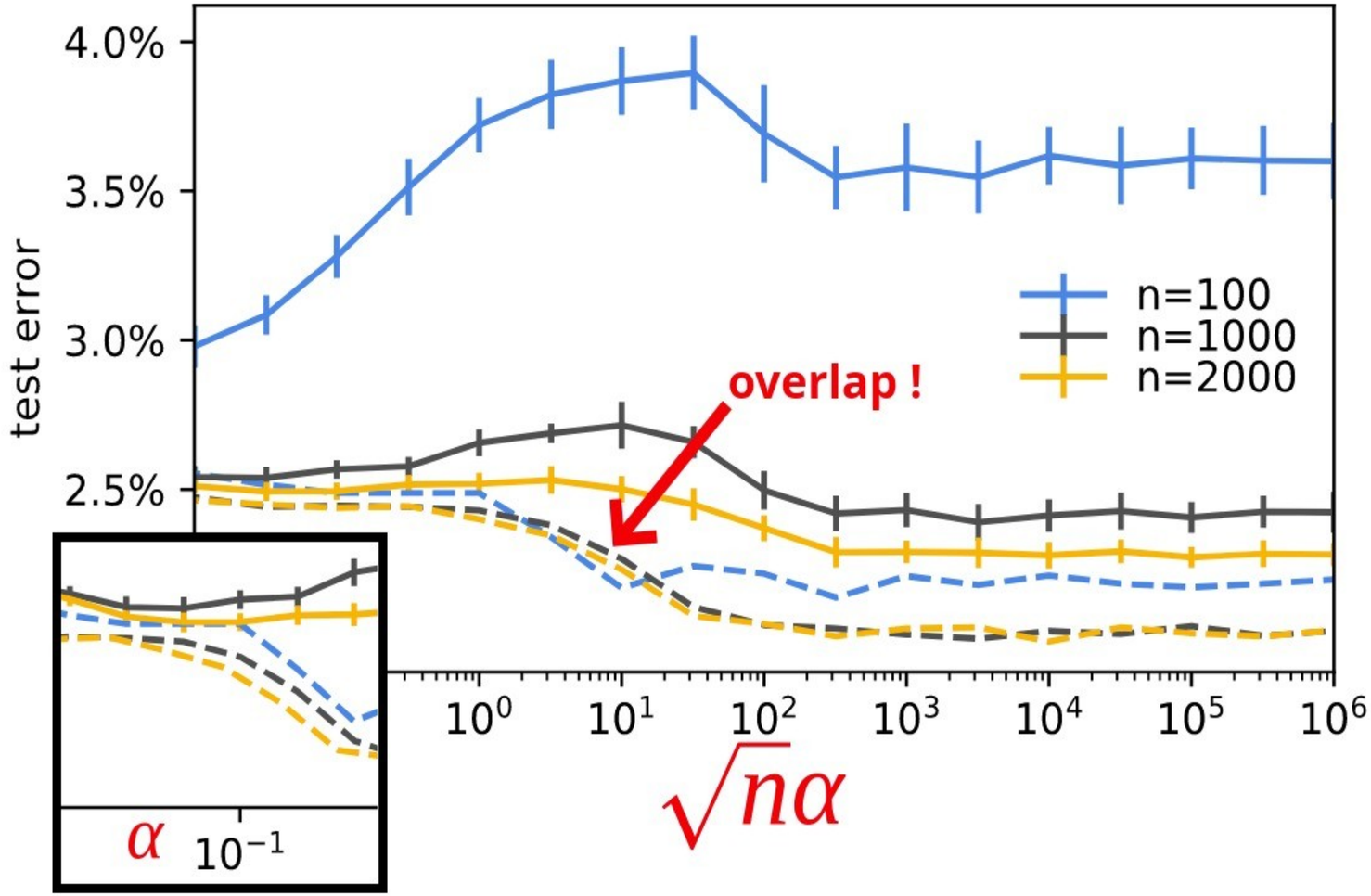


MNIST 10k parity, FC L=3, softplus, gradient flow with momentum

the ensemble average

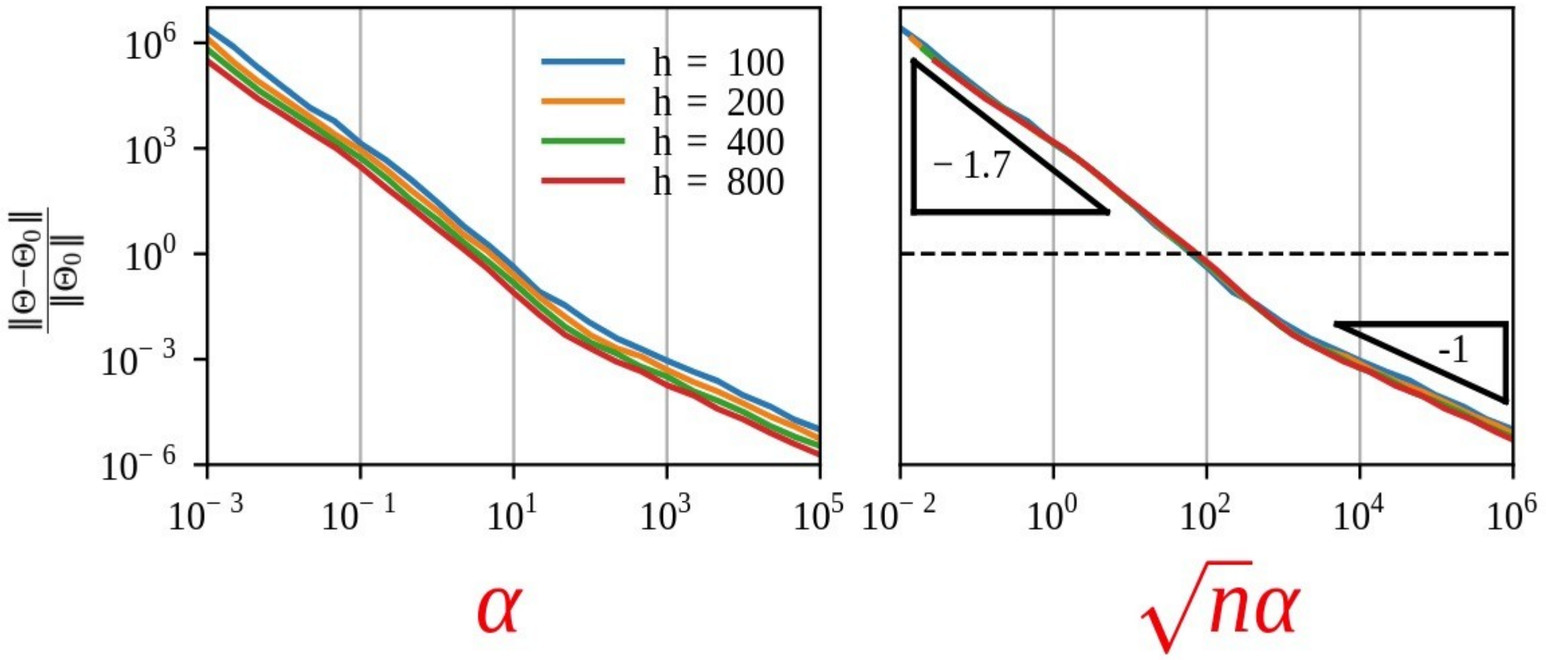
$$\bar{f}(x) = \int f(w(w_0), x)\, d\mu(w_0)$$

plot in function of $\sqrt{n\alpha}$ overlap the lines



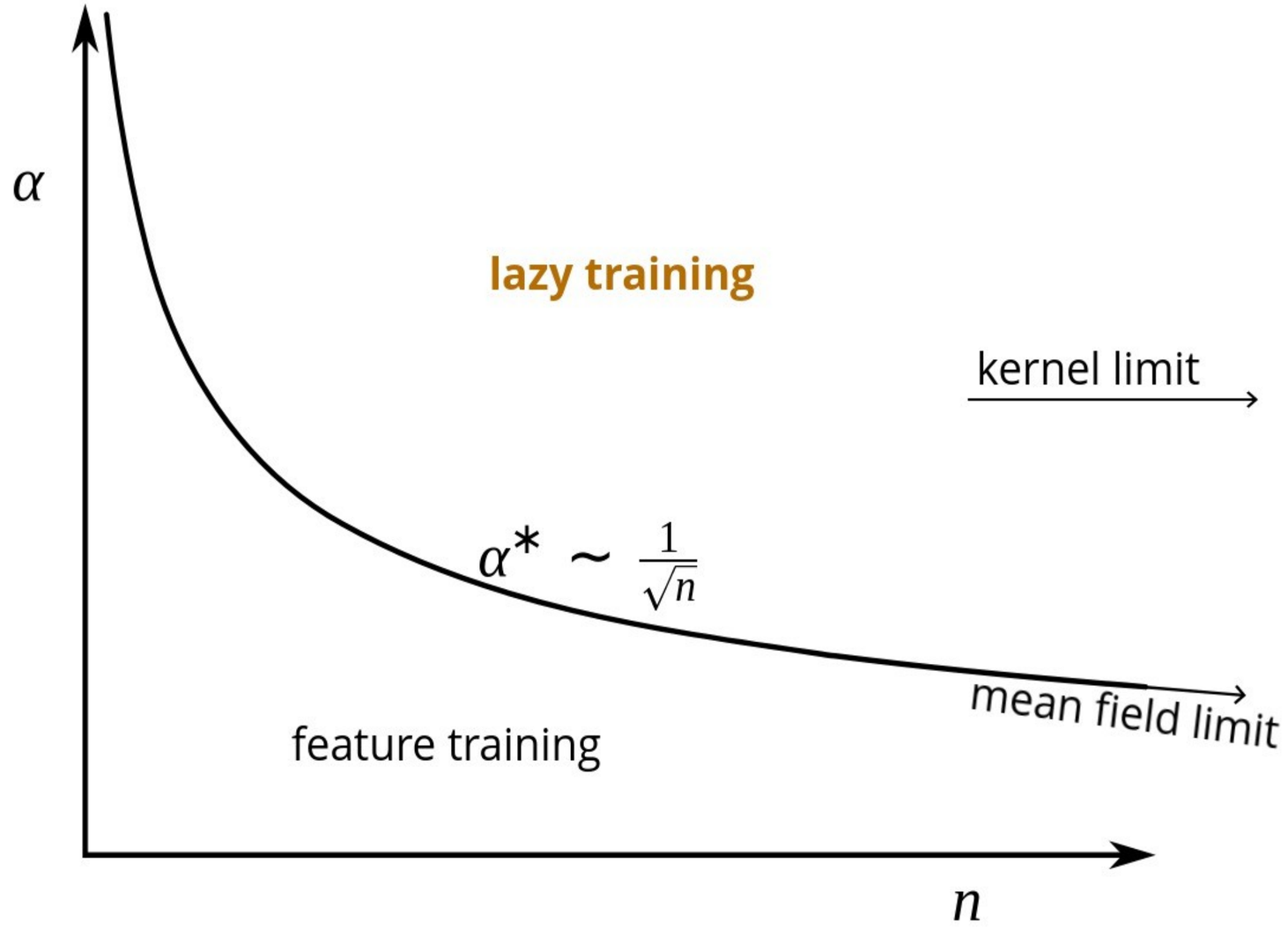
MNIST 10k parity, FC L=3, softplus, gradient flow with momentum

the kernel evolution displays two regimes

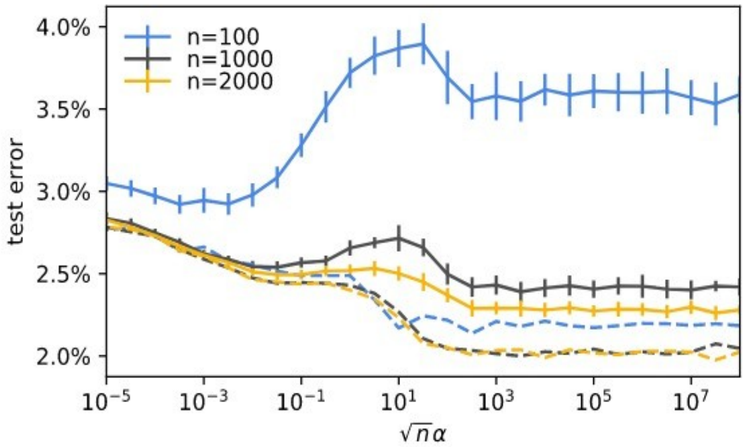


MNIST 10k parity, FC L=3, softplus, gradient flow with momentum

the phase space is split in two by α^* who decays with $\sqrt{n}$

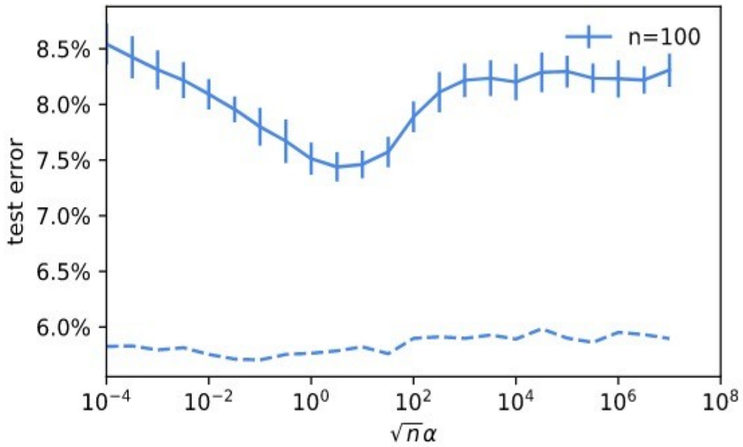


same for other datasets: the trends depends on the dataset



MNIST 10k

reduced to 2 classes

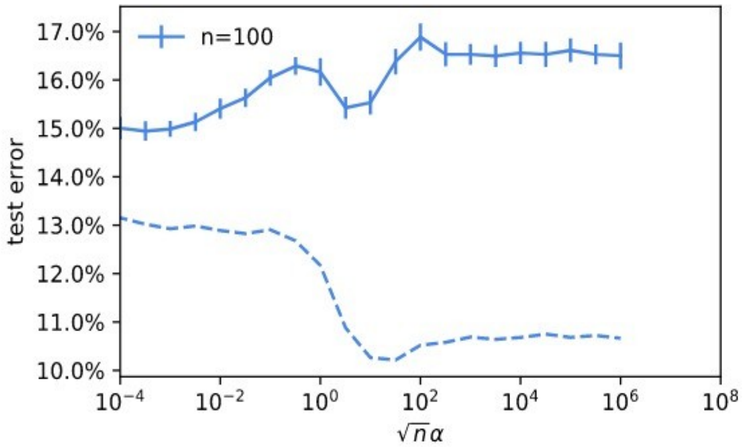


10PCA MNIST 10k

reduced to 2 classes

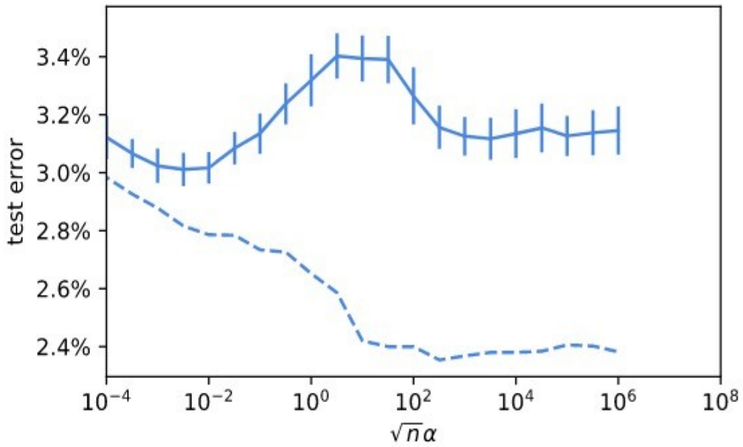
FC L=3, softplus, gradient flow with momentum

same for other datasets: the trends depends on the dataset



EMNIST 10k

reduced to 2 classes

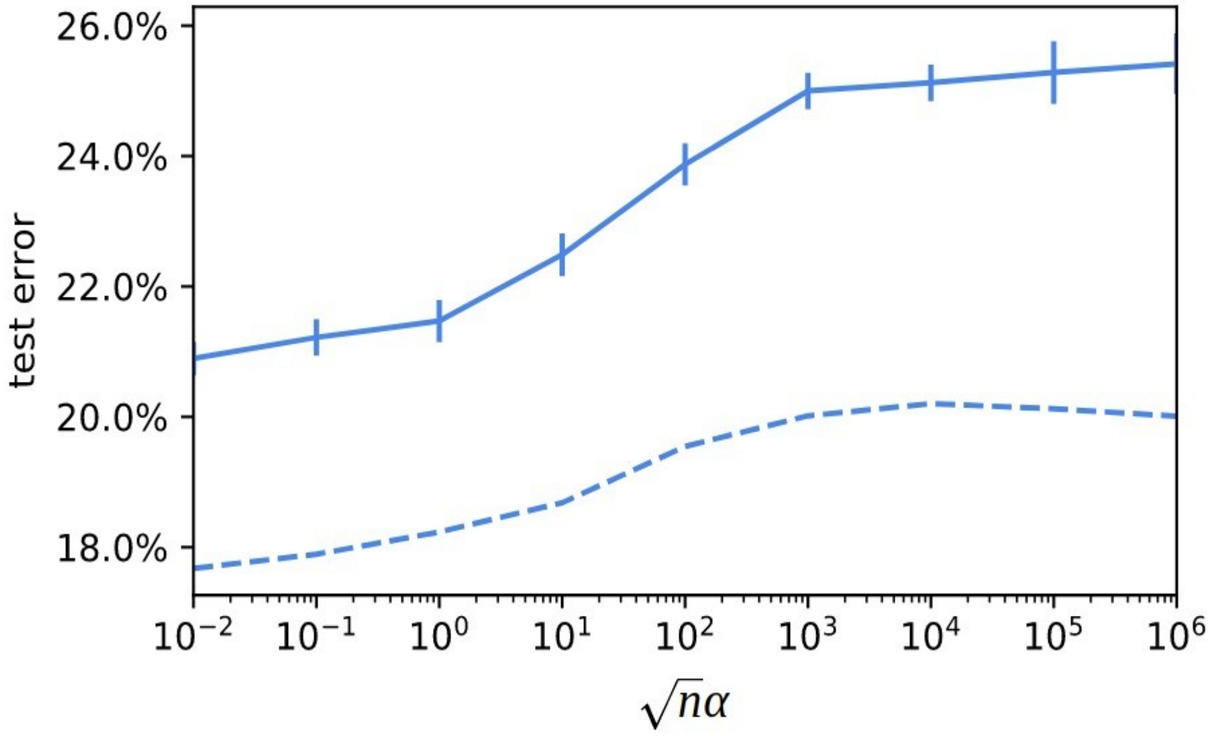


Fashion MNIST 10k

reduced to 2 classes

FC L=3, softplus, gradient flow with momentum

CNN: the tendency is inverted



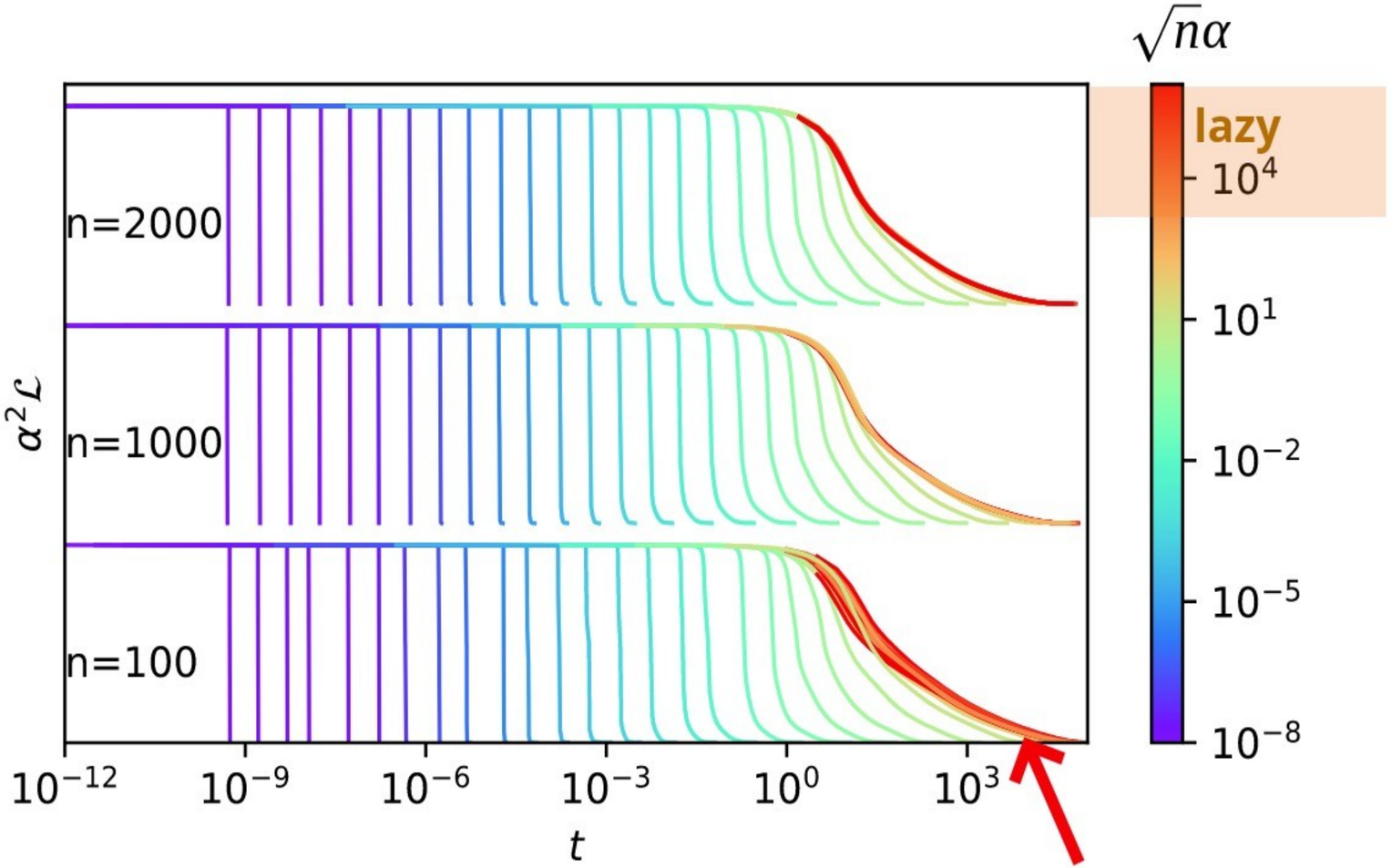
CIFAR10 10k

reduced to 2 classes

??

CNN SGD ADAM

how does the learning curves depends on n and α

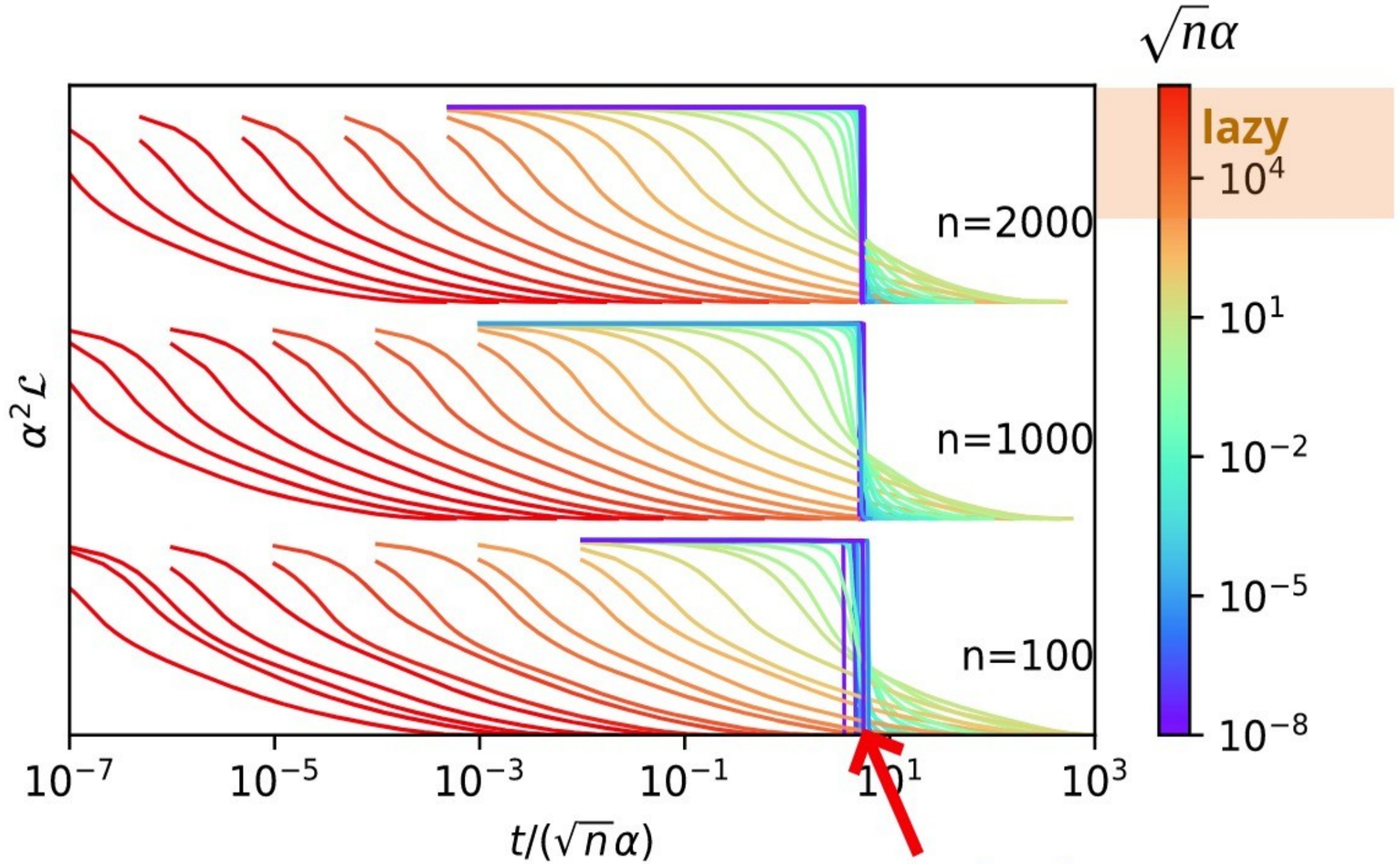


overlap !

same time in lazy

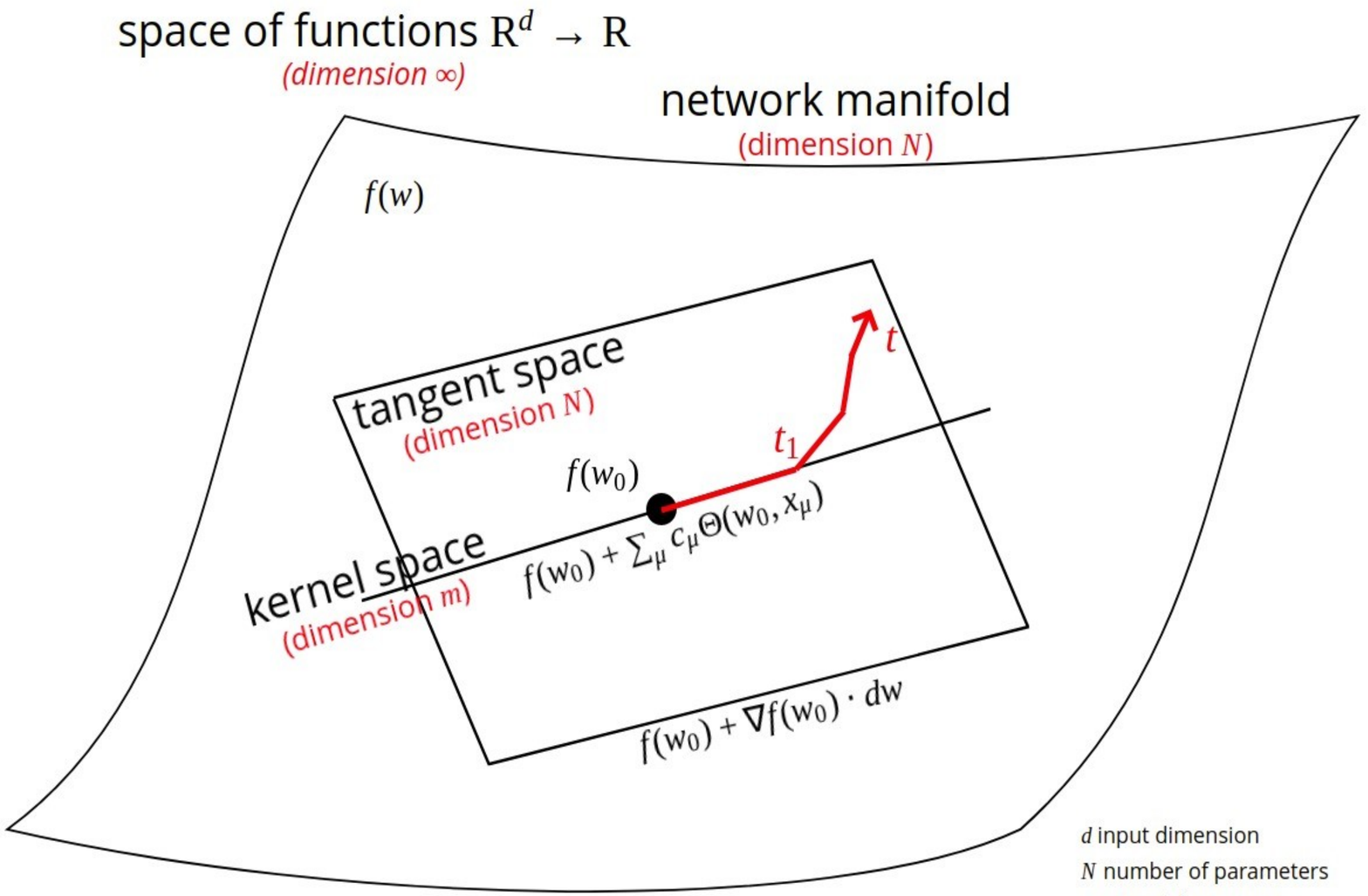
MNIST 10k parity, L=3, softplus, gradient flow with momentum

there is a characteristic time in the learning curves



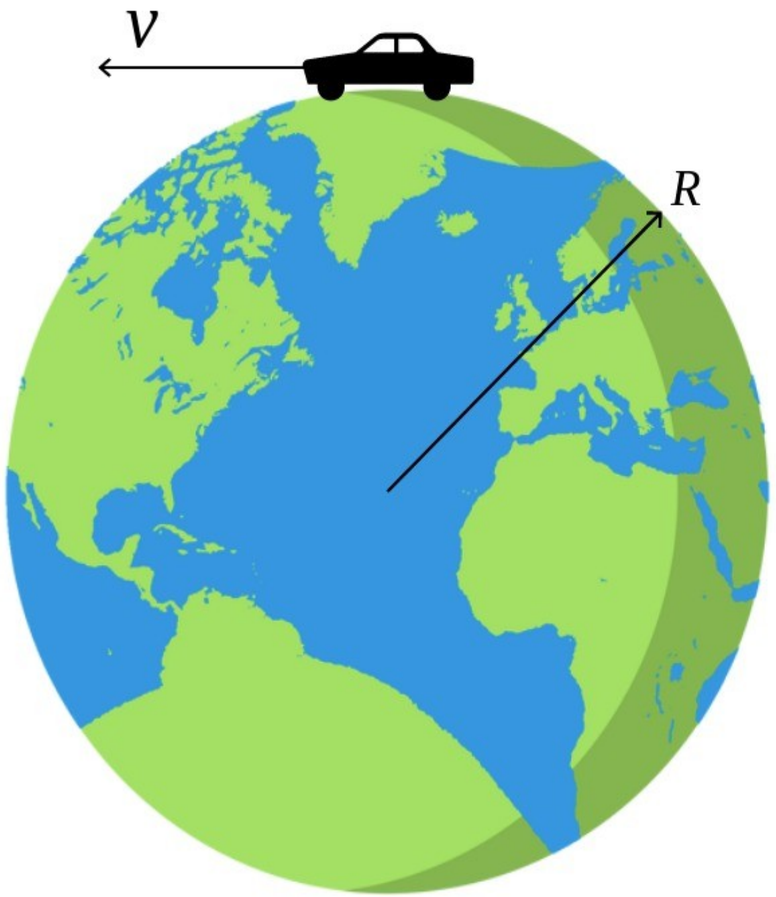
MNIST 10k parity, L=3, softplus, gradient flow with momentum

t_1 characterise the curvature of the network manifold



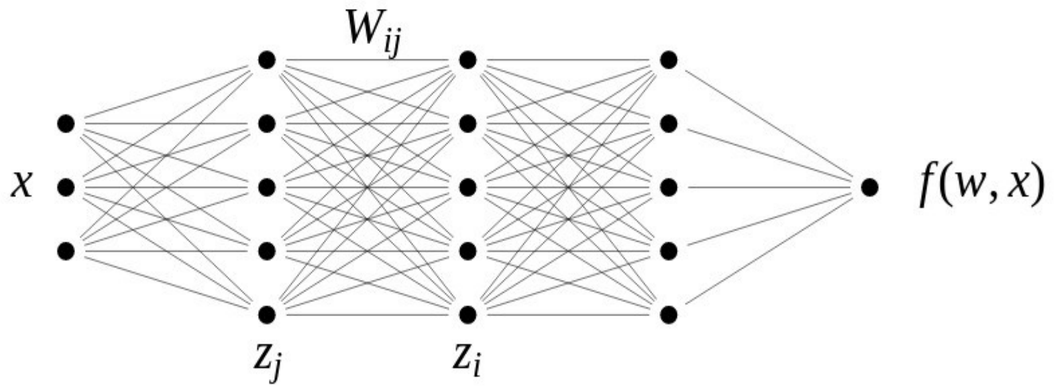
d input dimension
N number of parameters
m size of the trainset

t_1 is the time you need to drive to realize the earth is curved



$$t_1 \sim R/v$$

the rate of change of W determines when we leave the tangent space, aka $t_1 \sim \sqrt{n}\alpha$



$$z_i^{\ell+1} = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} W_{ij}^{\ell}\, \phi(z_j^{\ell})$$

W_{ij} and z_i are initialized ~ 1

$\dot{W}_{ij}$ and $\dot{z}_i$ at initialization $\Rightarrow t_1$

Upper bound: consider weight of last layer

$$\dot{W}^L = -\frac{\partial \mathrm{L}}{\partial W^L} = \mathrm{O}\left(\frac{1}{\alpha\sqrt{n}}\right)$$

$\Rightarrow t_1 \sim \sqrt{n}\alpha$

actually the correct scaling

$$\dot{W}^{\ell} = \mathrm{O}\left(\frac{1}{\alpha n}\right)$$

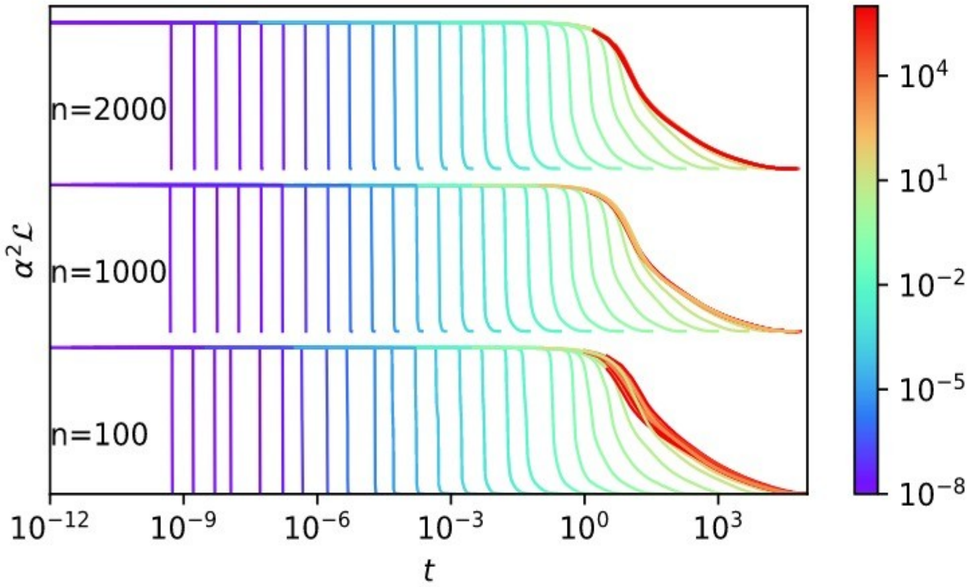
$$\dot{W}^{0} = \mathrm{O}\left(\frac{1}{\alpha\sqrt{n}}\right)$$

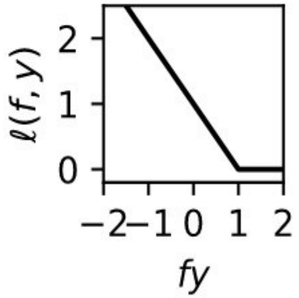
$$\dot{z} = \mathrm{O}\left(\frac{1}{\alpha\sqrt{n}}\right)$$

convergence time is of order 1 in lazy regime??

- $\mathrm{L}(w) = \frac{1}{\alpha^2 |\mathrm{D}|} \sum_{(x,y)\in \mathrm{D}} \ell\left(\alpha(f(w,x) - f(w_0,x)), y\right)$

$\rightsquigarrow$





convergence time is of order 1 in lazy regime??

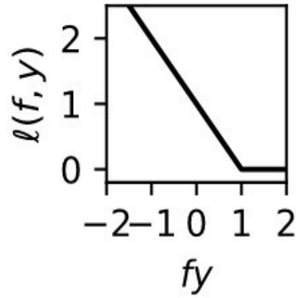
- $\mathrm{L}(w) = \frac{1}{\alpha^2 |\mathrm{D}|} \sum_{(x,y)\in \mathrm{D}} \ell\left(\alpha(f(w,x) - f(w_0,x)), y\right)$

  ??

- $\dot{f}(w,x) = \nabla_w f(w,x) \cdot \dot{w}$
- $= -\nabla_w f(w,x) \cdot \nabla_w \mathrm{L}$
- $\sim -\nabla_w f \cdot (\frac{\partial}{\partial f}\mathrm{L}\ \nabla_w f)$
- $\sim \frac{1}{\alpha}\Theta$

  ??

- "$\alpha \dot{f} t = 1$" $\Rightarrow$ "$t_{lazy} = \|\Theta\|^{-1}$"



when the dynamics stops before t_1 we are in the lazy regime

$$t_1 \sim \sqrt{n}\alpha$$

(time to exit the tangent space)

$$t_{lazy} \sim 1$$

(time to converge in the lazy regime)

$$?? \Longrightarrow ??$$

$$\alpha^* \sim \frac{1}{\sqrt{n}}$$

linearize the network with $f - f_0$ was not necessary

$$\alpha^* \sim 1/\sqrt{n}$$

then for large n

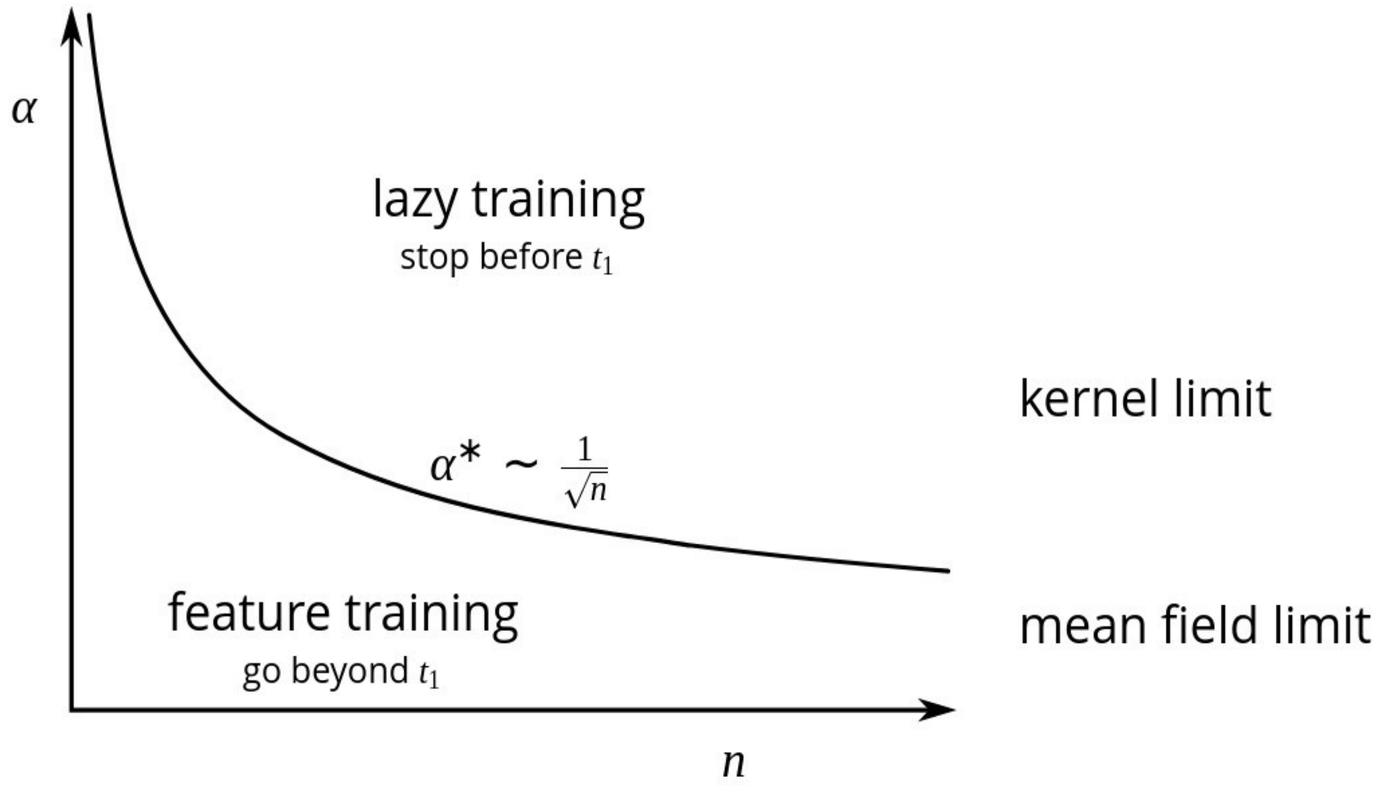
$$\Rightarrow$$

$$\alpha^* f(w_0, x) \ll 1$$

$$\Rightarrow$$

$$\alpha^*(f(w, x) - f(w_0, x)) \approx \alpha^* f(w, x)$$

for large n our conclusions should holds without this trick



time to leave the tangent space $t_1 \sim \sqrt{n}\alpha$

??=> time for learning features

arxiv.org/abs/1906.08034

github.com/mariogeiger/feature_lazy