Lazy vs hasty: linearization in deep networks impacts learning schedule based on example difficulty

Thomas George[1], Guillaume Lajoie[1], Aristide Baratin[2]

[1] Mila, Université de Montréal, McGill

[2] SAIT AI Lab, Montréal

TLDR

Non-linearly trained deep networks hasten towards easy examples much faster than linearly trained models

Goal

- insights into inductive bias of deep learning by studying training dynamics
- compare non-linear training dynamics to linearly-trained networks
- compare using easy/difficult examples

Setup

train function with lr rescaled by $\frac{1}{\alpha^2}$

$$f_\theta^\alpha(\mathbf{x}) := \alpha\left(f_\theta(\mathbf{x}) - f_{\theta_0}(\mathbf{x})\right)$$

α modulates regime:

- $\alpha = 1$ rich (feature learning) regime
- $\alpha \gg 1$ lazy (linear) regime
- ($\alpha < 1$ super adaptive regime)

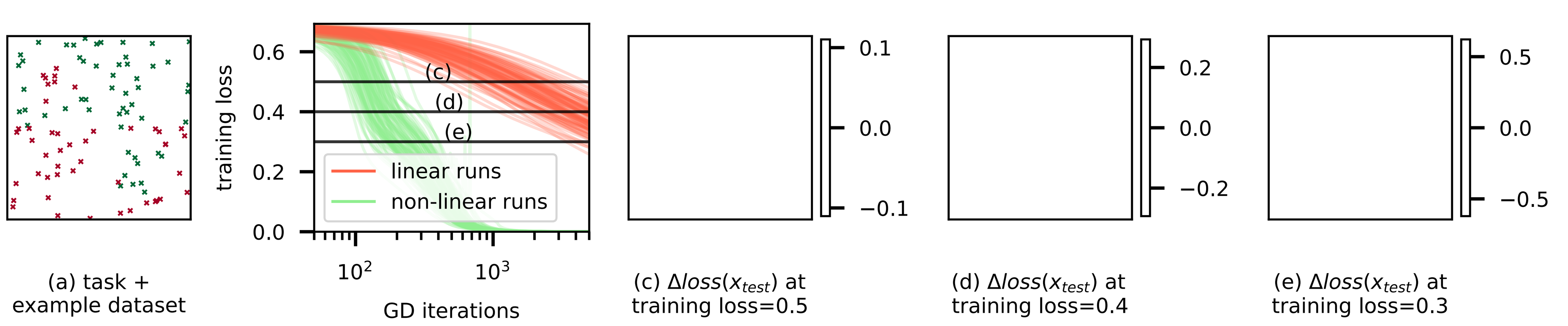
Background

Taylor series expansion: $f_\theta(\mathbf{x}) = f_{\theta_0}(\mathbf{x}) + (\theta - \theta_0)^\top \nabla_\theta f_{\theta_0}(\mathbf{x}) + \text{higher orders}$

NTK:
$K_\theta(\mathbf{x}, \mathbf{y}) = \langle \nabla_\theta f_\theta(\mathbf{x}), \nabla_\theta f_\theta(\mathbf{y}) \rangle$

Kernel alignment:
$\mathrm{KA}\left(\mathbf{K}^{(t)}, \mathbf{K}^{(0)}\right) := \frac{\mathrm{Tr}\left[\mathbf{K}^{(t)}\mathbf{K}^{(0)}\right]}{\left\|\mathbf{K}^{(t)}\right\|_F \left\|\mathbf{K}^{(0)}\right\|_F}$

Figure 1: Toy dataset



(a) task + example dataset
(c) $\Delta loss(x_{test})$ at training loss=0.5
(d) $\Delta loss(x_{test})$ at training loss=0.4
(e) $\Delta loss(x_{test})$ at training loss=0.3

Figure 2: C-scores

ResNet18 on CIFAR10

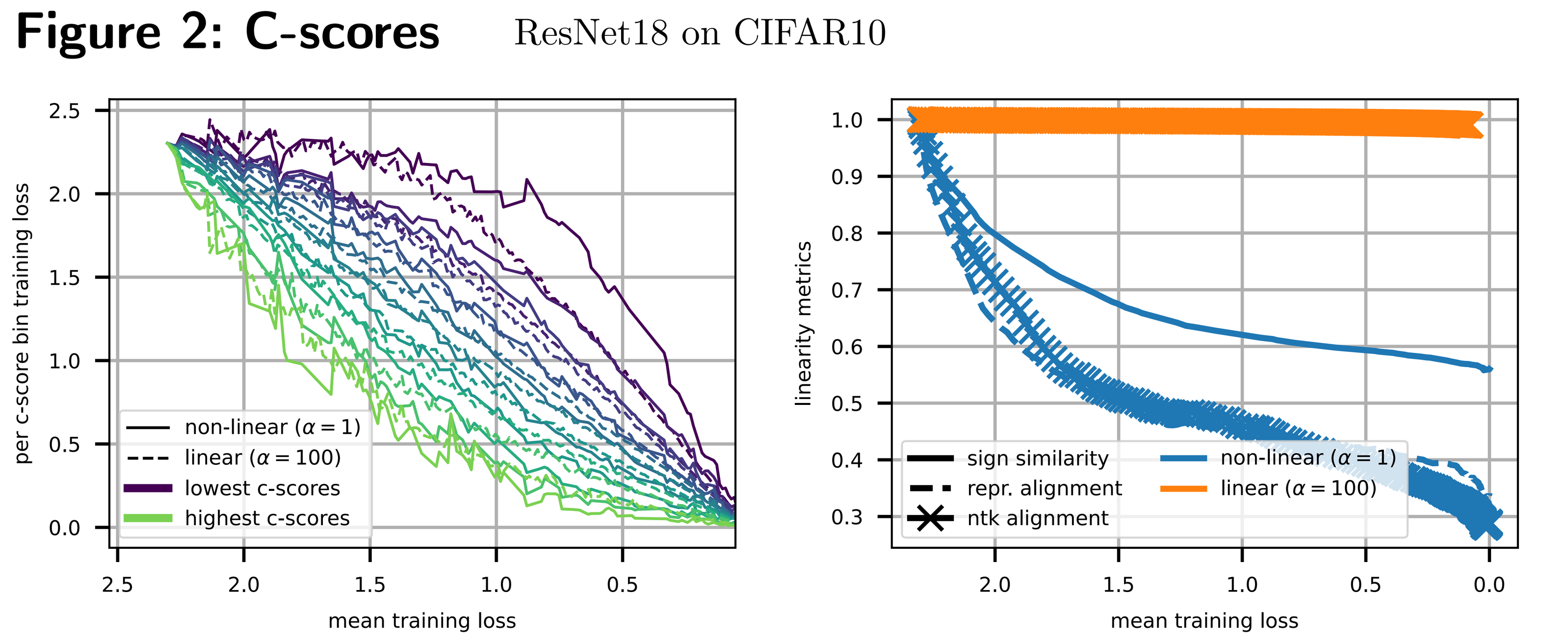


Figure 3: Noisy examples

ResNet18 on CIFAR10 with 15% noisy labels

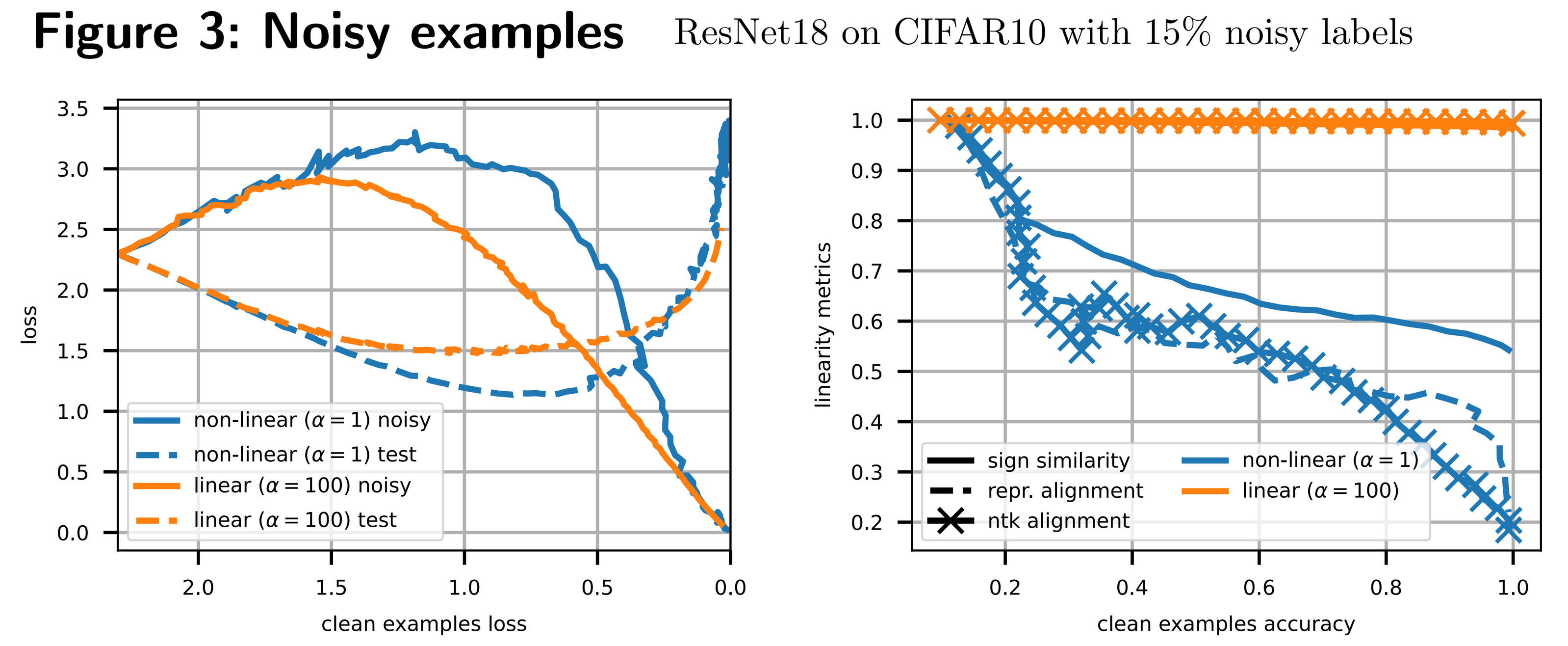
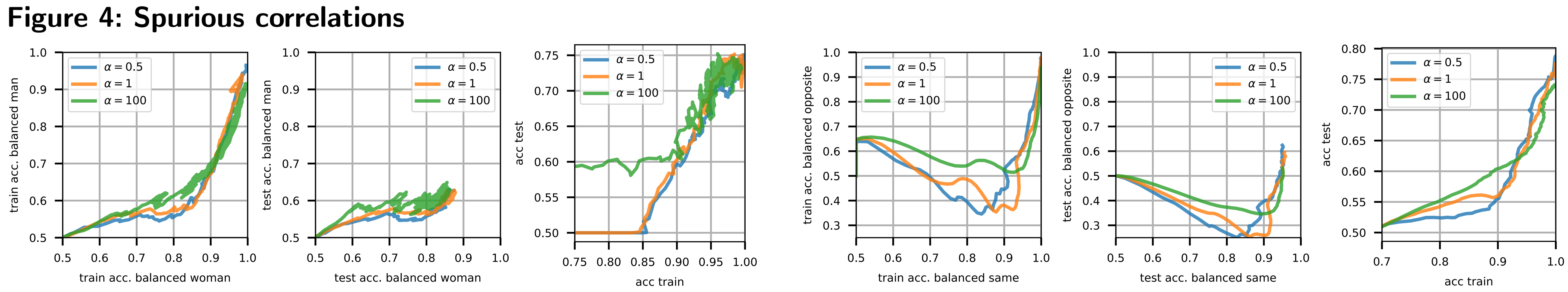


Figure 4: Spurious correlations



CelebA: ResNet18 trained with SGD+momentum

Waterbirds: Pre-trained ResNet18 finetuned with SGD+momentum

link to workshop paper: