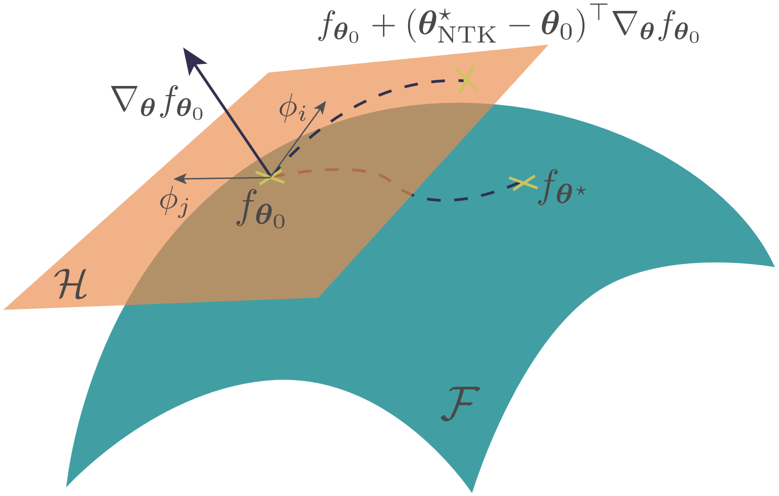
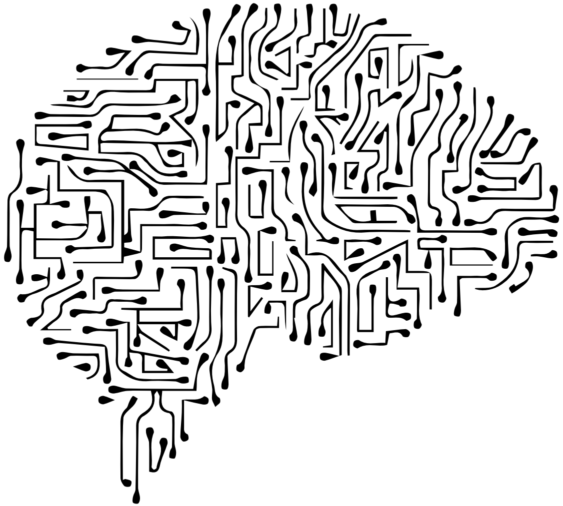
What can linearized neural networks *actually* say about generalization ?

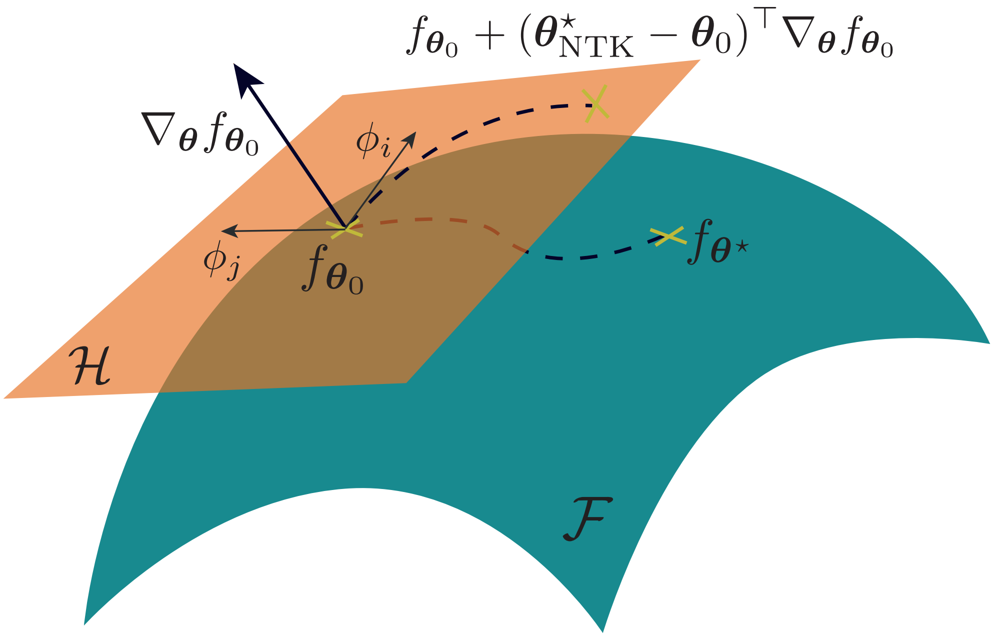
Guillermo Ortiz-Jiménez
Seyed-Mohsen Moosavi-Dezfooli
Pascal Frossard



EPFL
ETH zürich
NEURAL INFORMATION PROCESSING SYSTEMS

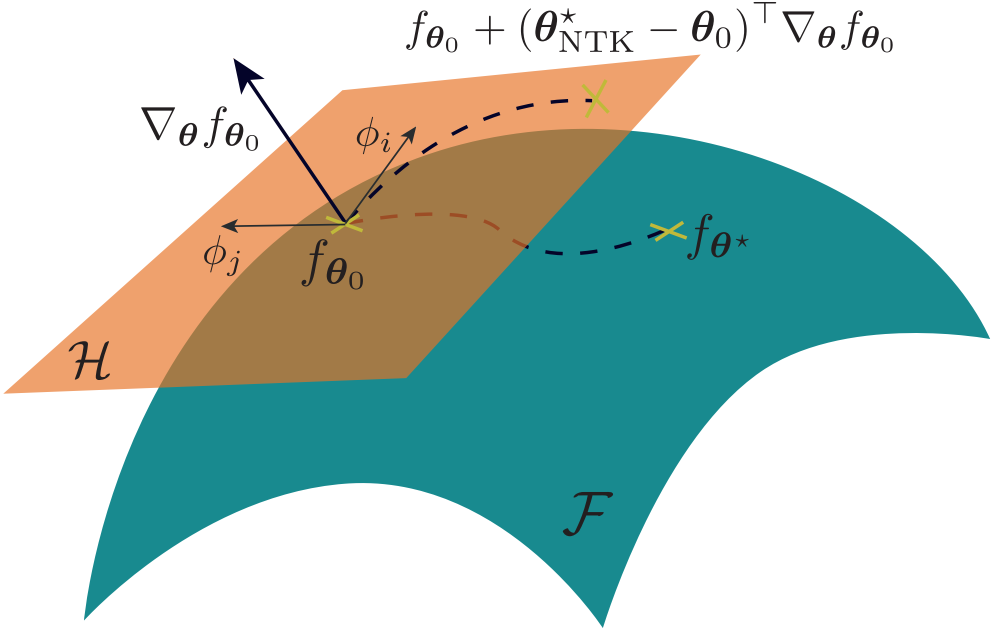


Deep Learning?



Neural Tangent Kernel.

(Jacot et al. 2019)



NTK.

(Jacot et al. 2019)

$$\underbrace{f_{\boldsymbol{\theta}}(\boldsymbol{x})}_{\text{Non-linear}} \approx \underbrace{f_{\boldsymbol{\theta}_0}(\boldsymbol{x}) + (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_0}(\boldsymbol{x})}_{\text{Linear}}$$

Neural tangent kernel (NTK):

$$\boldsymbol{\Theta}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_0}(\boldsymbol{x}_1), \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_0}(\boldsymbol{x}_2) \rangle$$

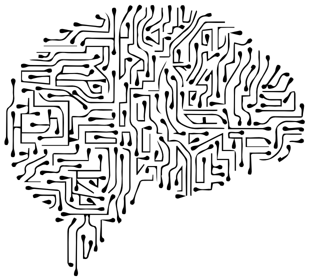
It defines a metric over functions:

$$\downarrow \|f\|^2_{\boldsymbol{\Theta}} = \sum_{j=1}^{\infty} \frac{1}{\underbrace{\lambda_j}_{\text{Eigenvalue}}} (\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\overbrace{\phi_j(\boldsymbol{x})}^{\text{Eigenfunction}} f(\boldsymbol{x})])$$

Proxy

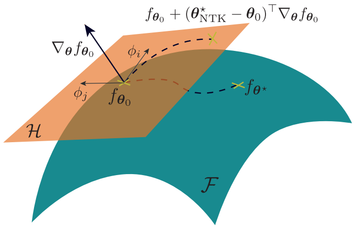
$$\uparrow \alpha(f) = \sum_{j=1}^{\infty} \lambda_j \, (\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\phi_j(\boldsymbol{x}) f(\boldsymbol{x})])$$

On many problems, provably...



Deep learning.

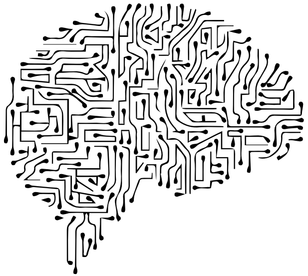
$\neq$



NTK.

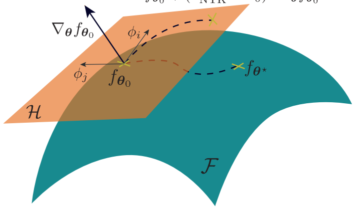
(Allen-Zhu & Li 2019, Ghorbani et al. 2020, Fort et al. 2020, Malach et al. 2021)

On many problems, provably...



Deep learning.

>>



NTK.

(Allen-Zhu & Li 2019, Ghorbani et al. 2020, Fort et al. 2020, Malach et al. 2021)

An NTK perspective on [insert topic]

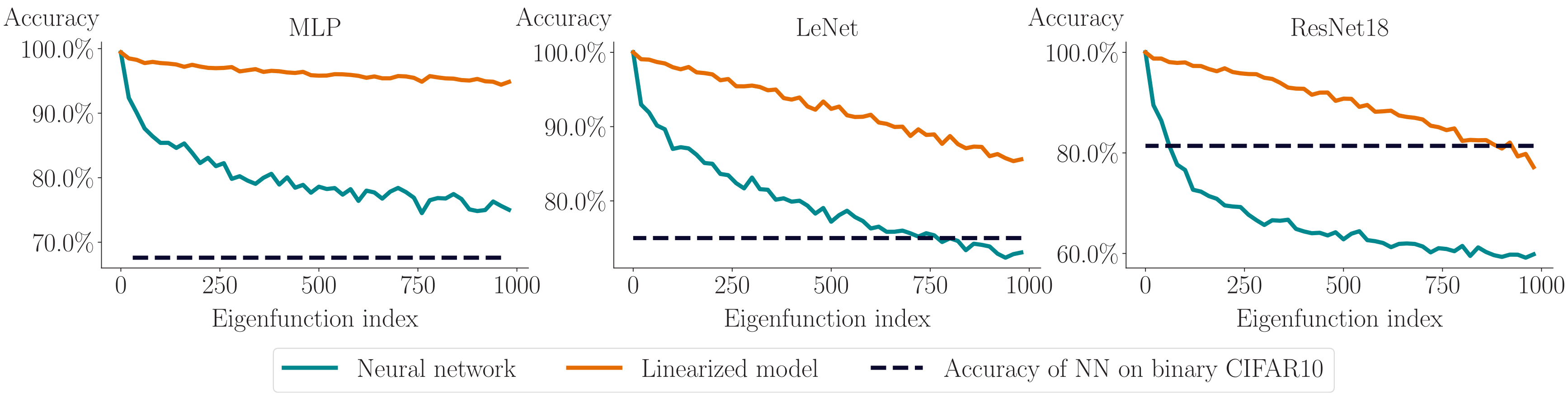
(Arora et al. 2020, Mobahi et al. 2020, Tancik et al 2020, Deshpande et al. 2021, Zancato et al. 2021, Gebhart et al. 2021, Bachman et al. 2021,, Maddox et al. 2021, and more.)

What can linearized neural networks *actually* say about generalization?

A new benchmark

Generate different tasks defined ***only*** by labels:

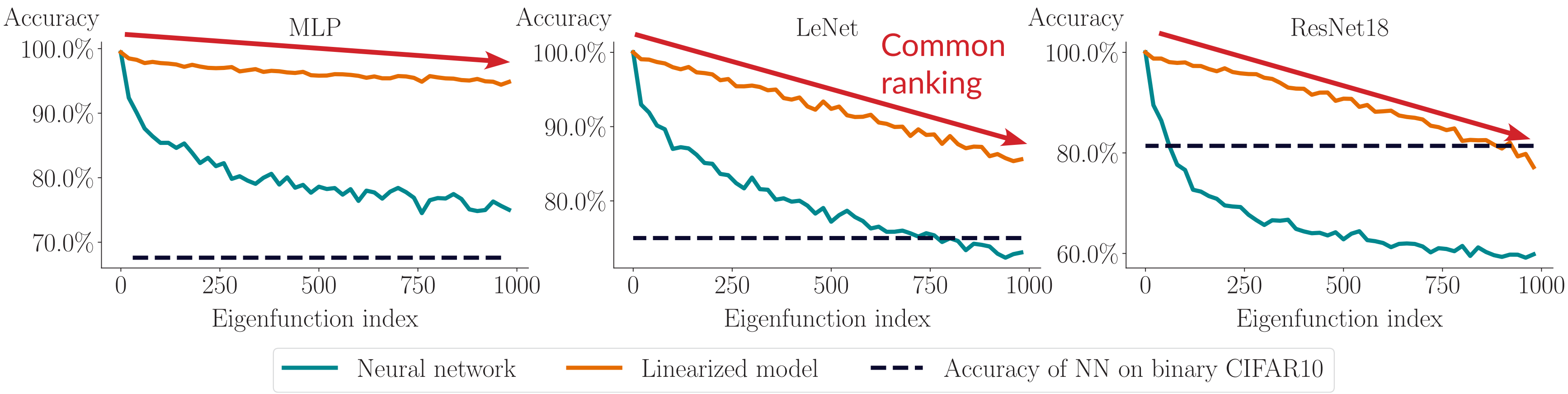
- Take CIFAR10 data
- Assign labels based on NTK: $\boldsymbol{x} \mapsto \mathrm{sign}(\phi_j(\boldsymbol{x}))$
- Train linear and non-linear models on each task.



A new benchmark

Generate different tasks defined ***only*** by labels:

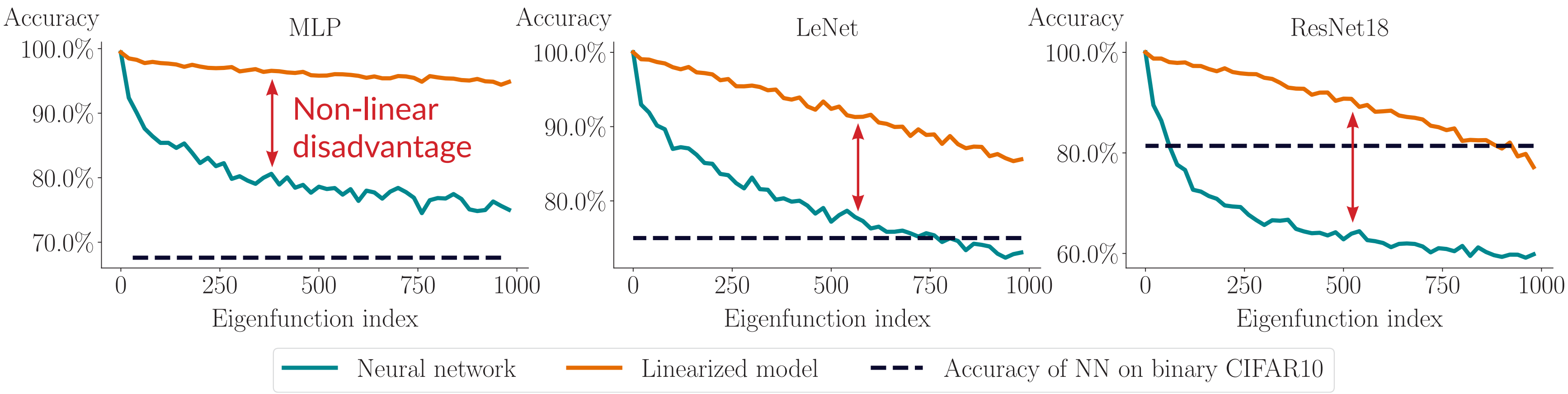
- Take CIFAR10 data
- Assign labels based on NTK: $\boldsymbol{x} \mapsto \mathrm{sign}(\phi_j(\boldsymbol{x}))$
- Train linear and non-linear models on each task.

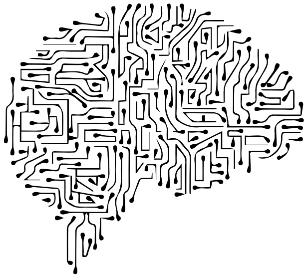


A new benchmark

Generate different tasks defined ***only*** by labels:

- Take CIFAR10 data
- Assign labels based on NTK: $\boldsymbol{x} \mapsto \mathrm{sign}(\phi_j(\boldsymbol{x}))$
- Train linear and non-linear models on each task.

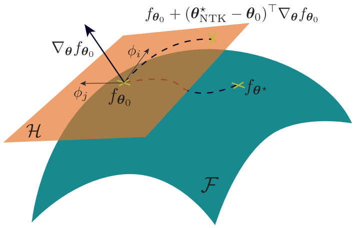




Deep learning.

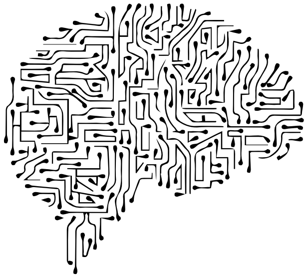
Ordering

$\approx$



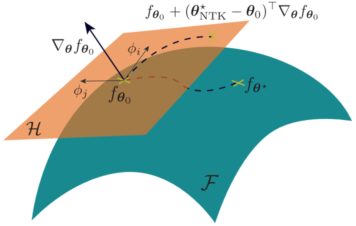
NTK.

On some problems...



Deep learning.

<

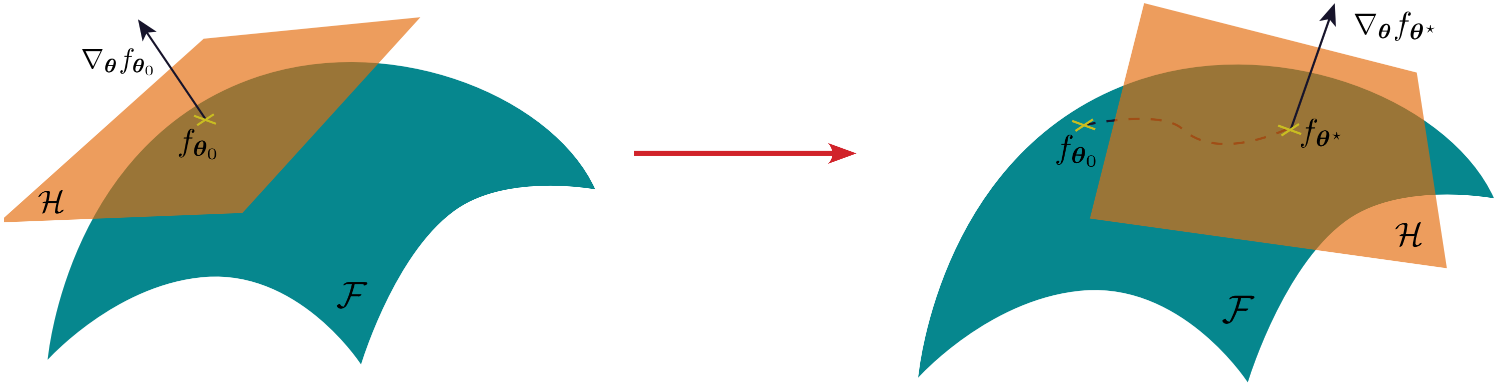


NTK.

What is the source of the non-linear (dis)advantage?

Neural network dynamics

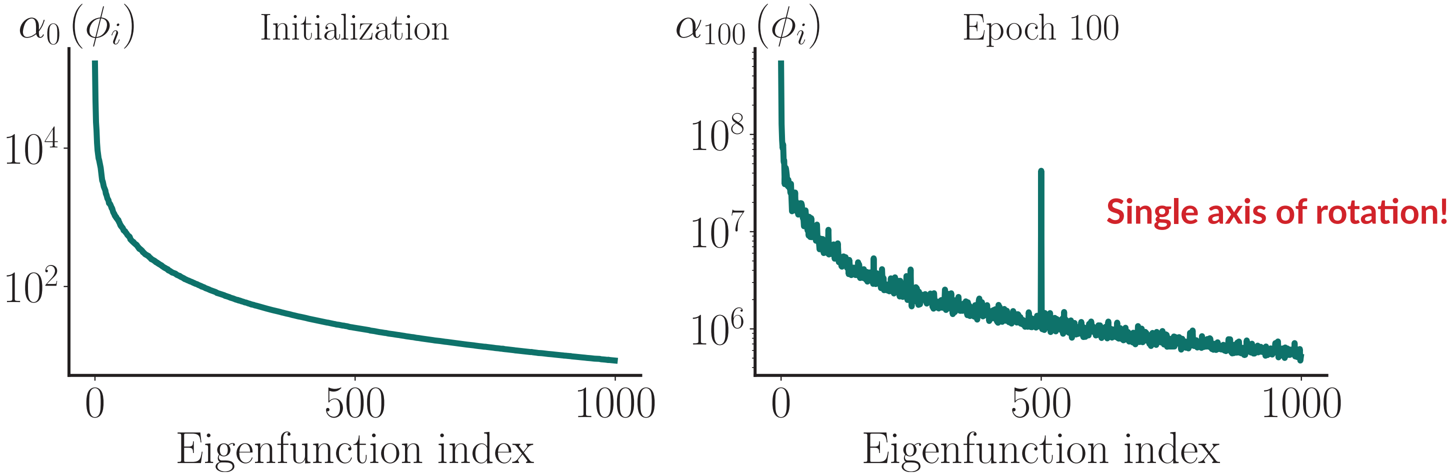
During training, $\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x})$ evolves and so does $\boldsymbol{\Theta}_t(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_1), \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_2) \rangle$



Neural network dynamics

During training, $\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x})$ evolves and so does $\boldsymbol{\Theta}_t(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_1), \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_2) \rangle$

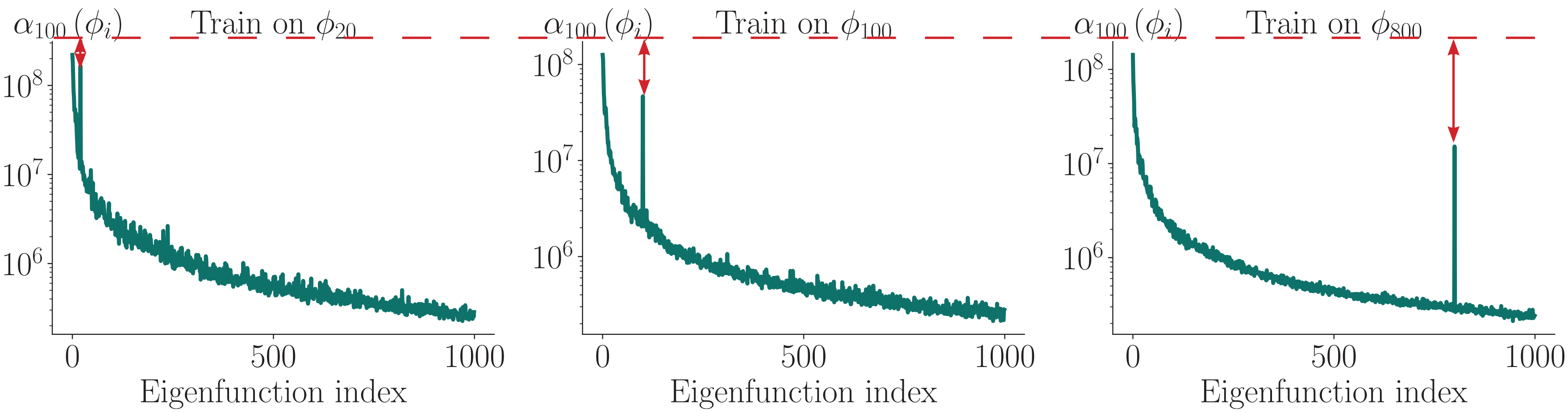
Alignment increases during training (Kopitkov & Indelman 2020, Paccolat et al. 2021, Baratin et al. 2021)

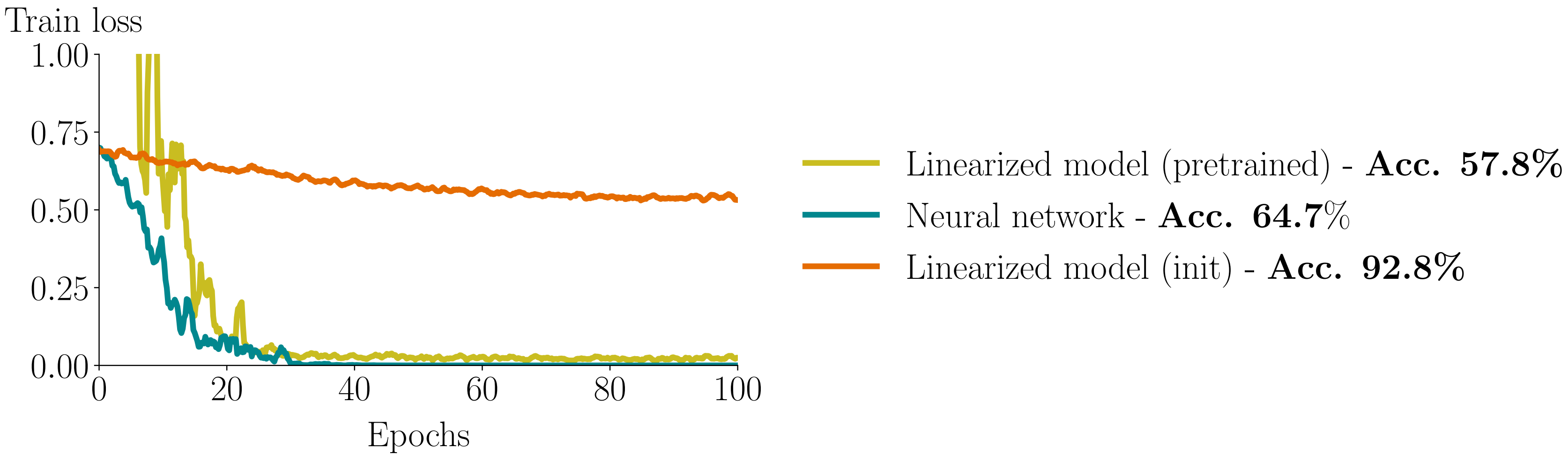


Neural network dynamics

During training, $\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x})$ evolves and so does $\boldsymbol{\Theta}_t(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_1), \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_2) \rangle$

Alignment increases during training (Kopitkov & Indelman 2020, Paccolat et al. 2021, Baratin et al. 2021)

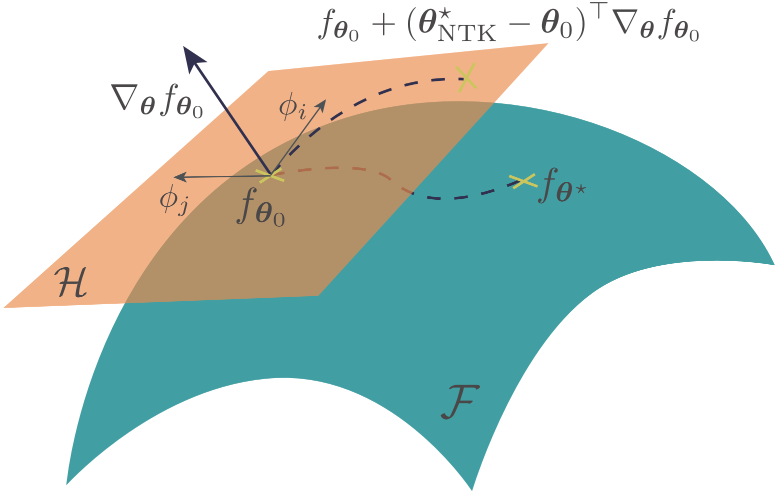




Train loss
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
100
Epochs
Linearized model (pretrained) - **Acc. 57.8%**
Neural network - **Acc. 64.7%**
Linearized model (init) - **Acc. 92.8%**

Concluding remarks

- Linear and non-linear models agree on relative difficulty of different tasks
- Neural networks are not always better than kernel methods.
- Kernel adaptation can make neural networks overfit.

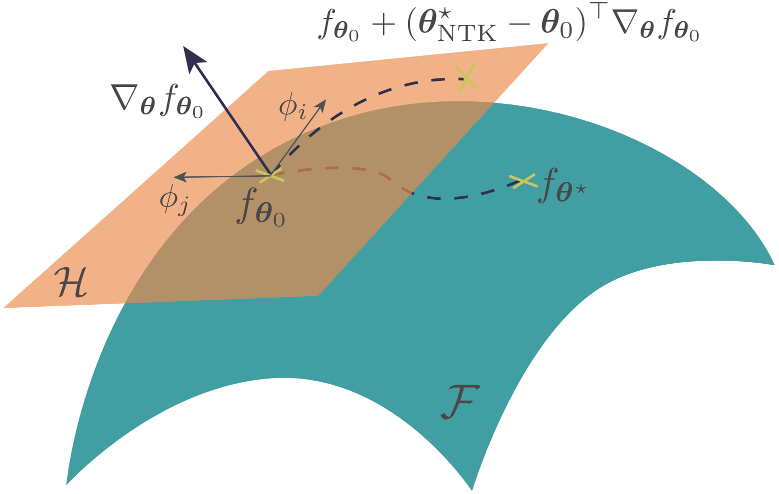


Concluding remarks

- Linear and non-linear models agree on relative difficulty of different tasks
- Neural networks are not always better than kernel methods.
- Kernel adaptation can make neural networks overfit.

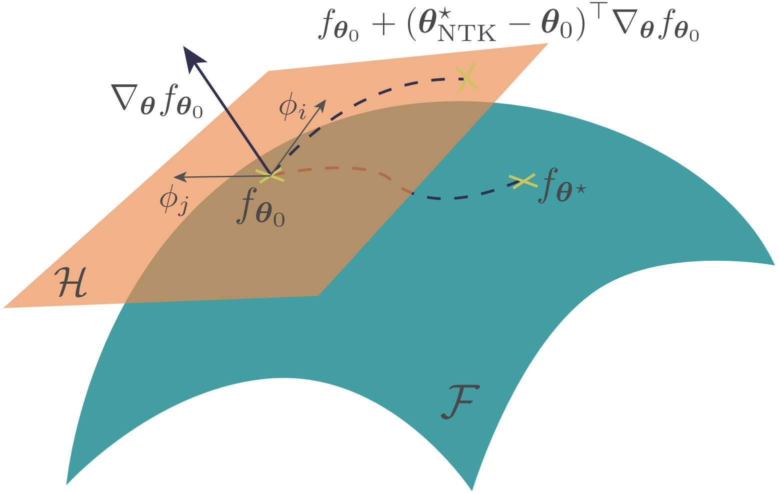
Also in the paper...

- We delve deeper into the observations,
- use alignment to predict inductive bias,
- study role of training set size in non-linear (dis)advantage.



What can linearized neural networks *actually* say about generalization ?

Guillermo Ortiz-Jiménez
Seyed-Mohsen Moosavi-Dezfooli
Pascal Frossard



EPFL ETH zürich NEURAL INFORMATION PROCESSING SYSTEMS