How a Student becomes a Teacher: learning and forgetting through Spectral methods

Lorenzo Giambagli[1,2], Lorenzo Buffoni[1], Lorenzo Chicchi[1], Duccio Fanelli[1]

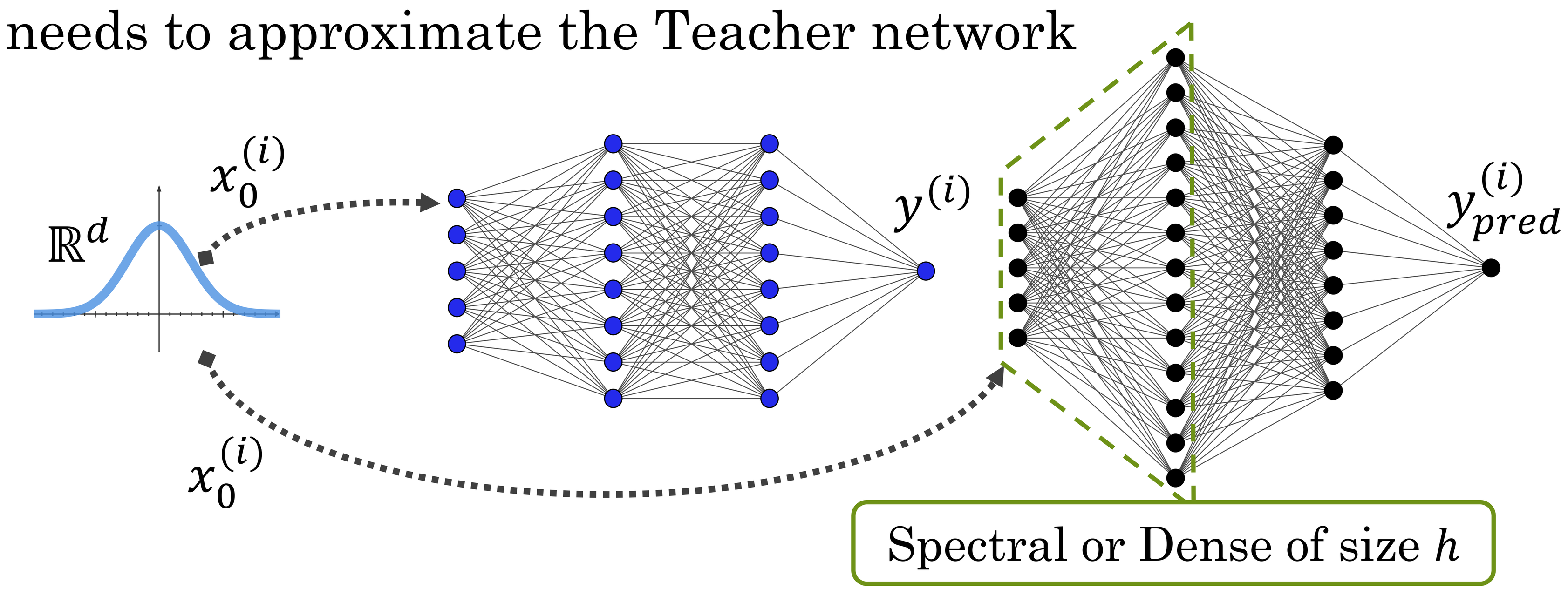
[1]CSDC, INFN, Department of Physics and Astronomy, University of Florence, Sesto Fiorentino, Italy

[2]naXys - Center for Complex Systems, Department of Mathematics, University of Namur, Namur, Belgium

Tensorflow and PyTorch
Spectral Layer + Node Pruning

Teacher Student framework

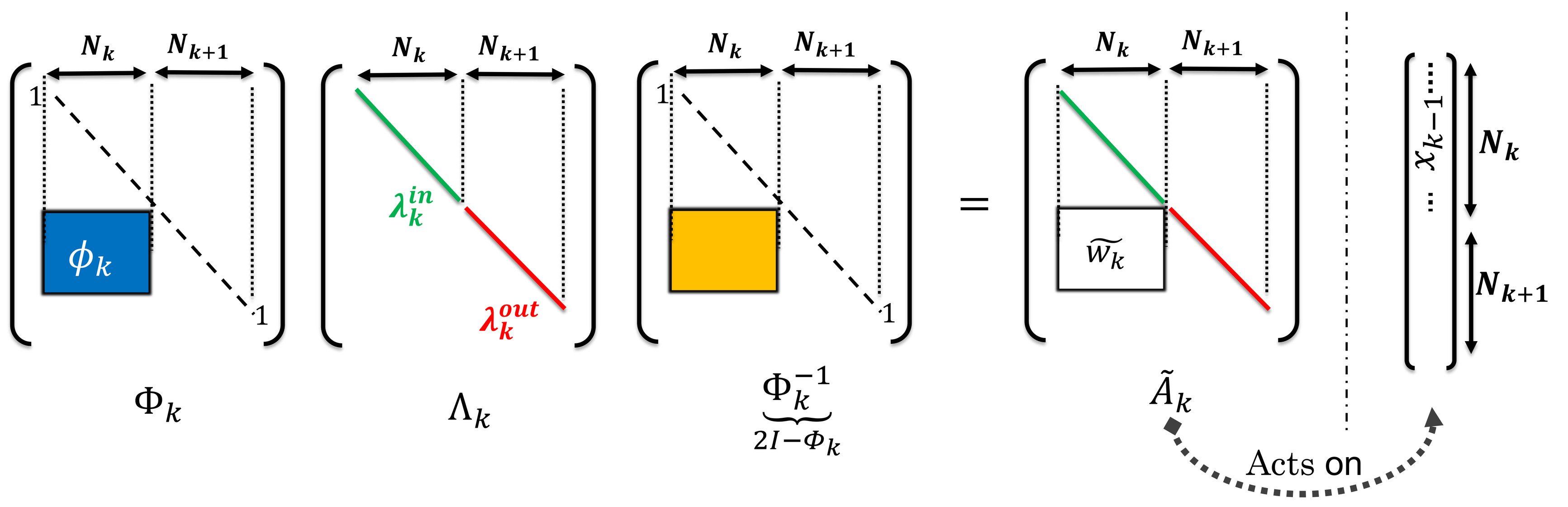
Widely used machine learning scheme. The Student network needs to approximate the Teacher network



Question: Can we know where the Teacher is inside the Student network?

Training in Spectral Domain

We **decompose** the **adjacency matrix** of the network into **eigenvalues** λ_k^{in} and λ_k^{out}, and **eigenvectors** ϕ_k, where k ranges across the layers of the neural network. The **learning** procedure is then **reframed** in terms of these global parameters, allowing for the simultaneous adjustment of multiple weights. Each transfer identifies two groups: *in*bound neurons (layer $k-1$) and *out*bound neurons (layer k).



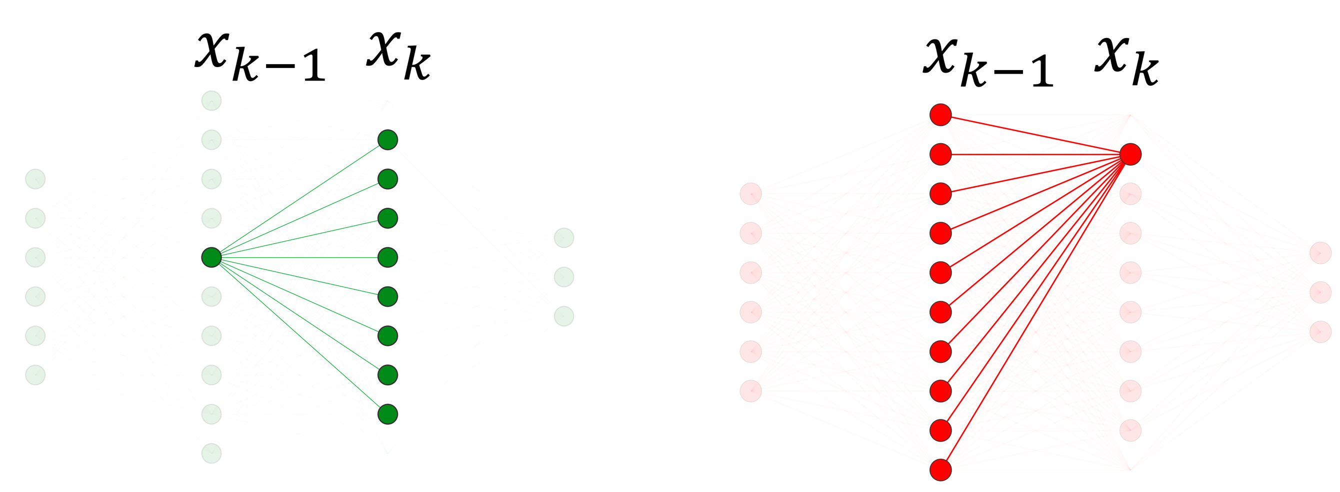
We can **parametrize** the **connection** with respect to the eigenvectors components and the eigenvalues obtaining a closed and simple formula.

Activity transfer from layer $k-1$ to k:

$$x_k = \widetilde{w}_k x_{k-1} = \sigma\left[\left(\phi_k \odot \boldsymbol{\lambda}_k^{in^T} - \boldsymbol{\lambda}_k^{out} \odot \phi_k\right) x_{k-1}\right]$$

In components, dropping index k on ϕ, λ

$$(x_k)_i = \sigma\left[\sum_j \left(\phi_{ij}\lambda_j^{in} - \lambda_i^{out}\phi_{ij}\right)(x_k)_j\right]$$



The **number of trainable $\boldsymbol{\lambda}$** are of the **same order** as the **neurons**. A much more efficient training is possible. In the following $\lambda^{in} = 0$ for simplicity. In this framework $\boldsymbol{\nabla}_{\lambda_k, \phi_k} Loss$

Spectral Regularization

Thanks to this novel approach, we can incorporate feature-oriented regularization. Furthermore, this relationship can be utilized to comprehend the **most significant nodes (features)** involved in information processing throughout the network.We define the following Loss function:

$$L = MSE(y, y_{pred}) + \Omega(\cdot)$$

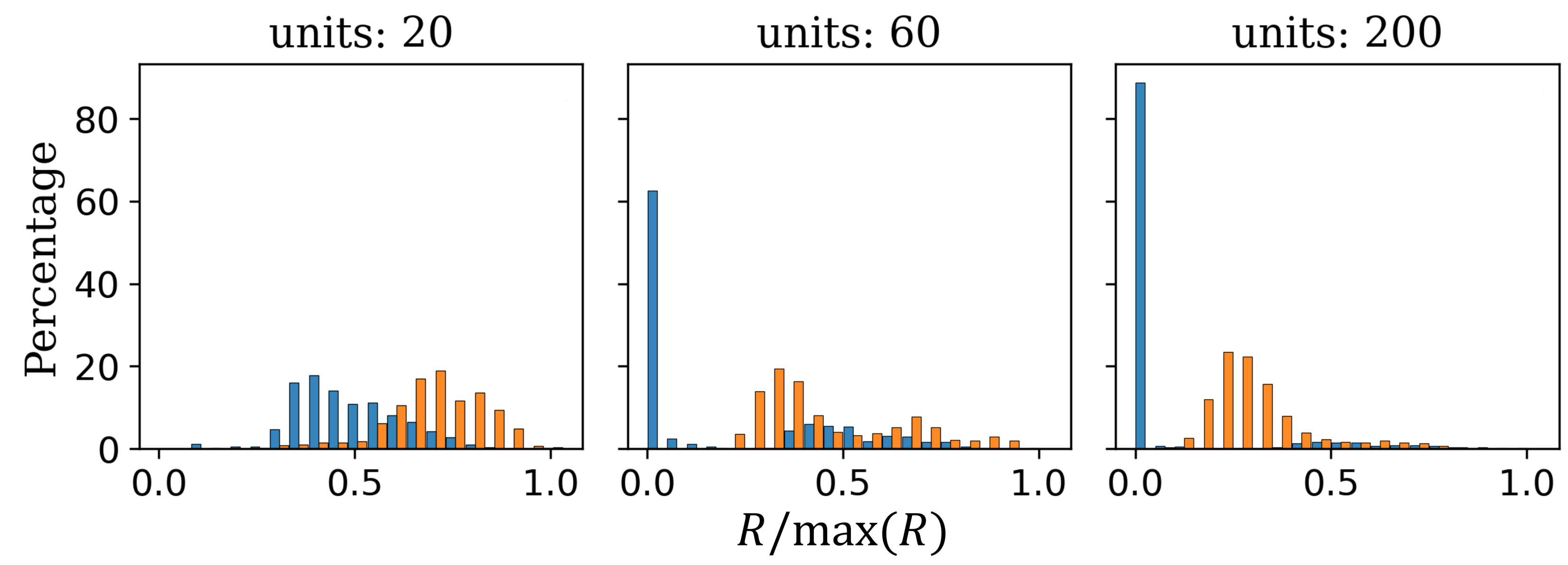
Where the regularization term Ω is:

- $\Omega(w) = \alpha_w L_2(w)$ if the layer is **Dense**
- $\Omega(\phi, \lambda) = \alpha_\phi L_2(\Phi) + \alpha_\lambda L_2(\lambda)$ if the layer is **Spectral**

Then the feature norm is extracted and compared. More specifically we have:

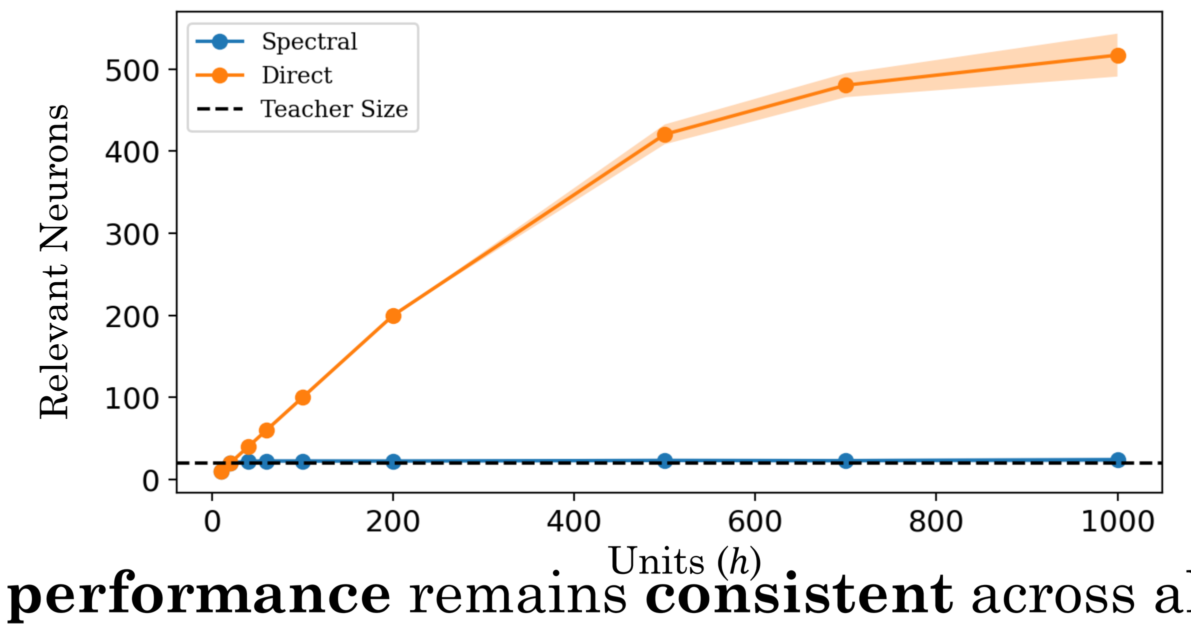
- $R_{dense}^j = \left(\sum_{k=1}^d w_{jk}^2\right)^{1/2}$ if the layer is Dense
- $R_{spectral}^j = \lambda_j^{out}\left(\sum_{k=1}^d \phi_{jk}^2\right)^{1/2}$ if the layer is Spectral

The **histogram** of R_{dense} (in orange) and $R_{spectral}$ (in blue) is **shown** for **different $\boldsymbol{h}$**. Remarkably, with the spectral regularization, a core of non-zero eigenvalues can be spotted as soon as $h > 20$, namely Teacher's dimension $h_{Teacher}$, whereas the large majority is basically zero. The same effect holds true for different Teacher sizes.

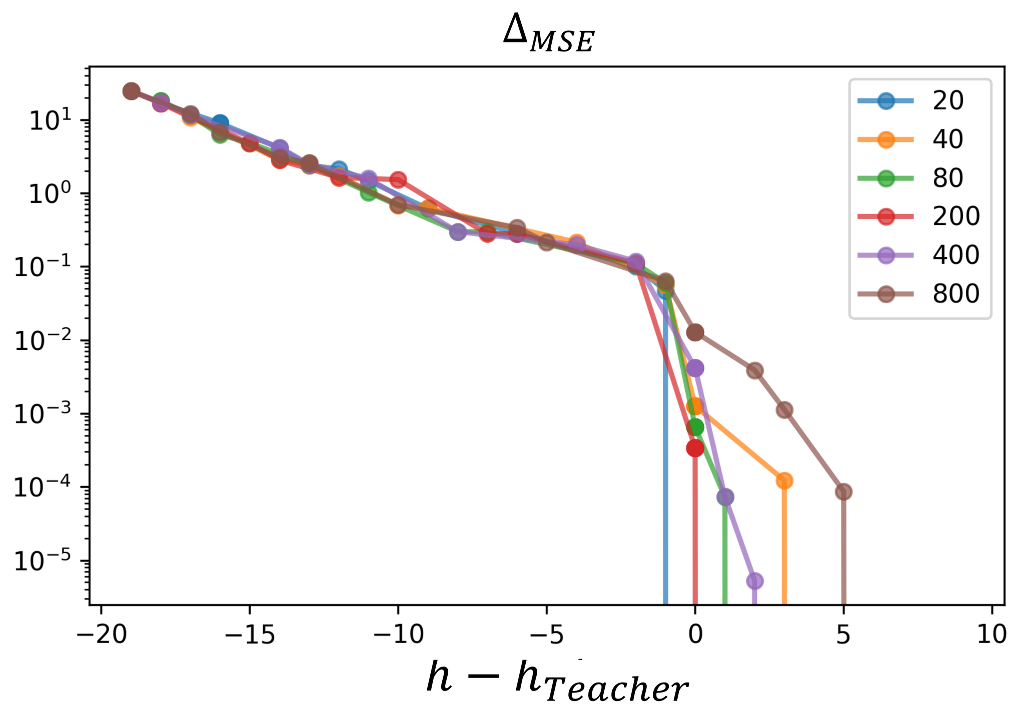


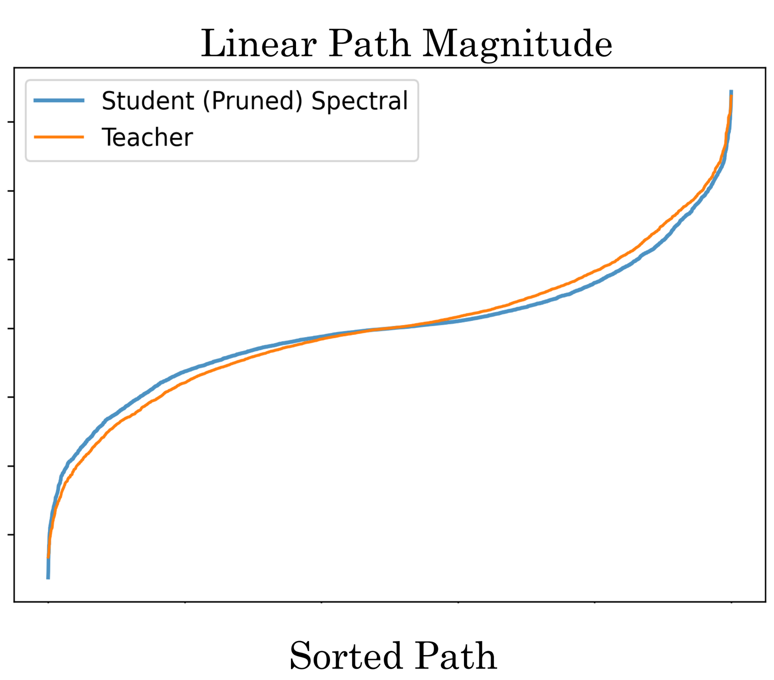
The Invariant Subnetwork

If we examine the size of the non-zero core across a wide range of h, we observe **distinct behaviours** depending on the type of regularization employed, whether it be Spectral or Weight Decay.



The **test performance** remains **consistent** across all models, irrespective of the type of layer employed. The **spectral** network is then **node pruned** based on the $R_{spectral}$ metric, which serves as a measure of feature relevance. When plotting the **variation** in mean squared error (**MSE**) with respect to the unpruned network, we observe a **phase transition-like** behavior. The trend of Δ_{MSE} remains consistent **regardless** of the **initial size** h, and the **critical point** occurs when the pruned network reaches **the same size** (complexity) as the **Teacher** network. Same results also with more realistic dataset (F-MNIST-MNIST-California Housing-CIFAR100 (with ResNet50 backbone)





References

- Giambagli, L., Buffoni, L., Carletti, T., Nocentini, W. & Fanelli, D. Machine Learning in Spectral Domain. *Nature Communications* 12, 1–9 (2021)
- Chicchi L., Giambagli L., Buffoni L., Carletti T., Ciavarella M., Fanelli D. Training of sparse and dense deep neural networks: Fewer parameters, same performance. Phys. Rev. E104 (2021)
- Buffoni, L., Civitelli, E., Giambagli, L., Chicchi, L., & Fanelli, D. Spectral Pruning of Fully Connected Layers *Scientific Reports* (2022)

Lorenzo Giambagli
Department of Physics, University of Florence (Italy)
Department of Mathematics & (naXys) Namur Institute for Complex Systems, Université de Namur (Belgium)
lorenzo.giambagli@unifi.it
lorenzo.giambagli@gmail.com