Learning Manifold and dimensionality reduction in Deep Learning and Geometric Deep Learning

Rita Fioresi, University of Bologna

September 5, 2024 Paris

CaLISTA COST Action

Join CaLISTA CA 21109!

https://site.unibo.it/calista/en

Action Chair: Rita Fioresi

- **Working group 1: Cap U. Vienna, Slovak U. Brno**
  Cartan Geometry and Representation Theory
- **Working group 2: Abenda U. Bologna, Tanzini Sissa**
  Integrable Systems and Supersymmetry
- **Working group 3: O Buachalla Charles U., Aschieri Uniupo**
  Noncommutative Geometry and Quantum Homogeneous Spaces
- **Working group 4: Angulo Mines Paris, Parton U. Pescara**
  Vision and Machine Learning
- **Working group 5: Lledo U. Valencia, Tekel U. Bratislava**
  Dissemination and Public Engagement

https://e-services.cost.eu/action/CA21109/working-groups/applications

CaLISTA COST Action Rules for REIMBURSEMENT

- Enroll in COST association www.cost.eu
- Associate to CaLISTA CA 21109
- Click on the official invitation of Irena Vanatova GHR
- Sign the presence sheets and the contract
- Upload reimbursement documents within **15 DAYS**

COST Action CaLISTA Events 2024-2025

- July 1-5, 2024. Training School "Integrable System", Lisbon.
- Sept 2-5, 2024. Training School "Geometry Informed Machine Learning", Paris.
- Sept 25-26, 2024. Worshop on "Lie and Quantum GLq", Zagreb.
- October 4, 2024. Workshop "Women and Nonbinary Researchers of CaLISTA", Bratislava.
- June 2-5, 2025. Workshop "Integrable Systems", Leeds
- June 17, 2025. Workshop "Geometry and Machine Learning", Toulouse.
- June 30-July 1, 2025. Workshop "Quantum Groups", Cambridge.
- mid September 2025. General Meeting of CaLISTA, Corfu'.

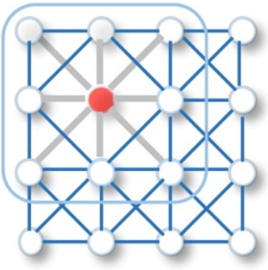
Plan of the Talk

1. Deep Learning and Geometric Deep Learning
2. Information Geometry
3. Fisher matrix and Data information matrix
4. Foliation in Deep Learning (Joint work with Tron)
5. Thermodynamic inspired parameter pruning in (Geometric) Deep Learning (Joint work with Lapenna, Faglioni)

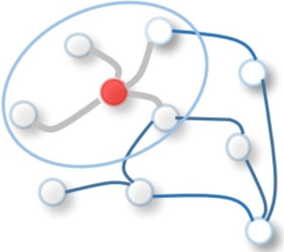
1. Deep Learning and Geometric Deep Learning

Introduction to (Geometric) Deep Learning

- Deep Learning: Convolutional Neural Networks (CNN)
- Deep Learning for Supervised Classification Tasks e.g. classification of images
- Geometric Deep Learning: CNN on non Euclidean domains, i.e. data naturally organized as a graph(s).



(a) 2D Convolution on an image



(b) Graph Convolution

Ingredients for (Geometric) Deep Learning

- **Score function**: it is a function of the weights w (es. linear classifier) It gives a *score* for a data x and weights w: e.g. $s(x, w) = \sum w_{ij} x_j$.
- **Loss function**: measures error (L_i datum i loss, y_i correct label)

$$L_i = -log \frac{e^{f_{y_i}}}{\sum_j e^{f_j}} = -f_{y_i} + log \sum_j e^{f_j}, \qquad L = \sum_i L_i$$

- **Optimizer**: for weights update “minimizes” the Loss

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \nabla L_{\mathrm{stoc}}, \qquad \nabla L_{\mathrm{stoc}} = \sum_{i=1}^{32} \nabla L_{\mathrm{rand(i)}}$$

Training

Divide the dataset (ex. CIFAR10):
80% Data for **training**
10% Data for **validation**
10% Data for **test** (ONCE)

1. **Learning**: determine weights **parameters**
2. **Validation**: determine net structure.
   Example: choose loss function, number of layers, learning rate
   Goal: find best hyperparameters.
3. **Test**: once at the end.

Accuracy: percentage of accurate predictions on tests set.

2. Information Geometry

Information Geometry

Information Geometry: studies geometrical structures on manifolds in the parameter space (space of probability distributions) and the data domain.

Amari, S.-I. *Natural gradient works efficiently in learning. Neural computation*, 10(2):251-276, 1998.

Amari Loss: $l(x, w) = -\log(p(y|x, w))$ (Loss function)

Loss function: $L(x, w) = \mathbb{E}_{y\sim q}[l(x, w)]$ (Empirical loss)

$$L(x, w) = \mathbb{E}_{y\sim q}[-\log(p(y|x, w))] = \mathrm{KL}(q(y|x)||p(y|x, w)) + \mathrm{constant}$$

$p(y|x, w) = (p_i(y|x, w))_{i=1,\dots,C}$: discrete probability distribution of data x
$q(y|x)$: mass discrete probability distribution.
C: classification labels y.
w: parameters.

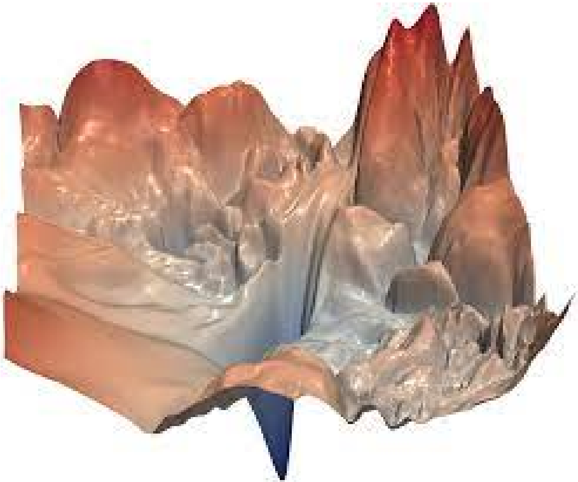
Loss Function

The empirical Loss function as expected value of the Amari Loss:

$$
\begin{aligned}
L(x,w) &= \mathbb{E}_{y\sim q}[-\log(p(y|x,w))] = \\
&= \sum_{i=1}^{C} q_i(y|x)\log\frac{q_i(y|x)}{p_i(y|x,w)} - \sum_{i=1}^{C} q_i(y|x)\log q_i(y|x) = \\
&= \mathrm{KL}(q(y|x)||p(y|x,w)) - \sum_{i=1}^{C} q_i(y|x)\log q_i(y|x). \qquad (1)
\end{aligned}
$$

The Kullback-Leibler divergence measures the “difference” between the two probability distributions the “empirical distribution” p and the “true distribution” q.

Loss Landscape



3. **The Fisher matrix F and the data information matrix G**

The Fisher matrix F and the data information matrix G

$$F(x,w) = \mathbb{E}_{y\sim p}[\nabla_w \log p(y|x,w)\cdot(\nabla_w \log p(y|x,w))^T]$$

$$G(x,w) = \mathbb{E}_{y\sim p}[\nabla_x \log p(y|x,w)\cdot(\nabla_x \log p(y|x,w))^T].$$

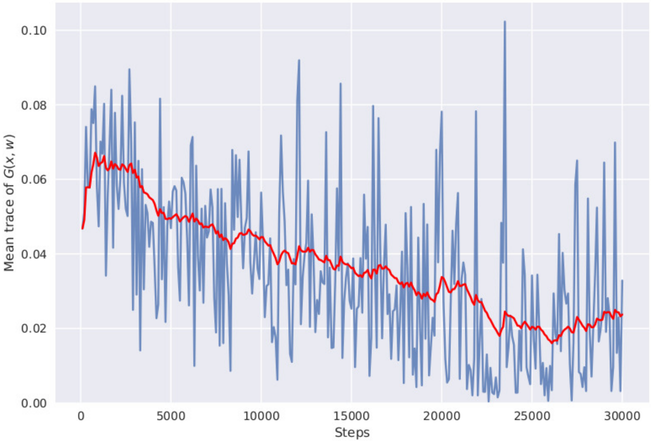
Key Facts:

$$\mathrm{KL}(p(y|x,w+\delta w)||p(y|x,w)) \cong \frac{1}{2}(\delta w)^T F(x,w)(\delta w) + \mathcal{O}(||\delta w||^3)$$

$$\mathrm{KL}(p(y|x+\delta x,w)||p(y|x,w)) \cong \frac{1}{2}(\delta x)^T G(x,w)(\delta x) + \mathcal{O}(||\delta x||^3)$$

The Fisher matrix F provides a natural metric on the **parameter space** during dynamics of the stochastic gradient descent.
The data information matrix G provides a **natural metric on the data domain**.

The data information matrix G during optimization



This is why we do not want a fully trained model: the information is lost at equilibrium!

Properties of the Fisher matrix F and data information matrix G

1. $F(x, w)$ and $G(x, w)$ is a positive semidefinite symmetric matrix.
2. $\ker F(x, w) = (\mathrm{span}_{i=1,\dots,C}\{\nabla_w \log p_i(y|x, w)\})^{\perp}$;
3. $\ker G(x, w) = (\mathrm{span}_{i=1,\dots,C}\{\nabla_x \log p_i(y|x, w)\})^{\perp}$.
4. $\mathrm{rank}\ F(x, w) < C, \quad \mathrm{rank}\ G(x, w) < C.$

Dataset	$G(x, w)$ size	$\mathrm{rank}\ G(x, w)$ bound
MNIST	784	10
CIFAR-10	3072	10
CIFAR-100	3072	100
ImageNet	150528	1000

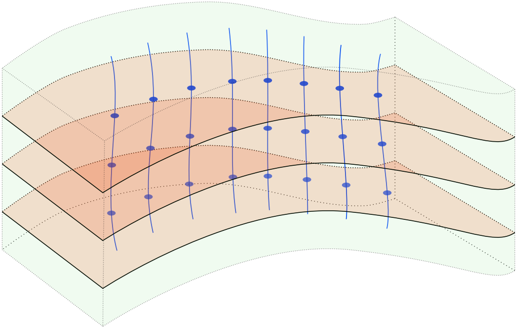
C: is the number of classes for our classification task

The Geometric Structure of Data: Distributions

Two orthogonal distributions emerge spontaneously:

$$\mathcal{D} = \mathrm{Im}\, G(x,w) = \mathrm{span}_{i=1,\dots,C}\{\nabla_x \log p_i(y|x,w)\}$$

$$\mathcal{D}^{\perp} = \ker G(x,w) = (\mathrm{span}_{i=1,\dots,C}\{\nabla_x \log p_i(y|x,w)\})^{\perp}$$



4. Foliations on the data domain

The Geometric Structure of Data: Foliations

Deep Learning and classification tasks:

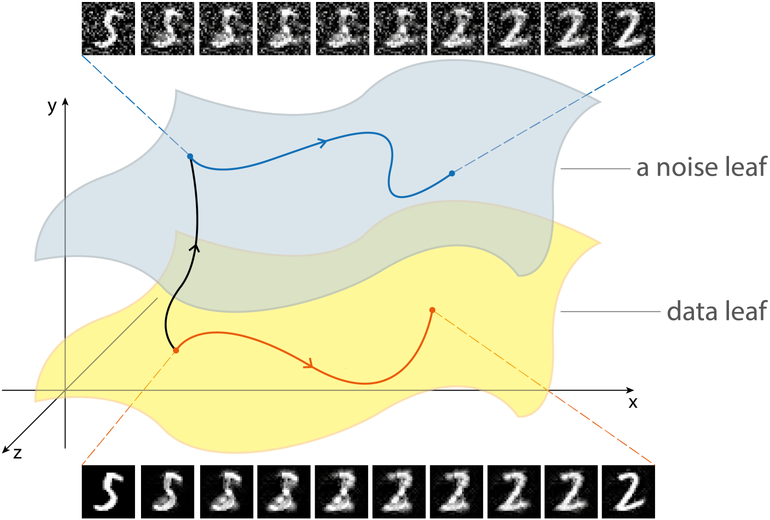
- Data occupies a domain in $\mathbb{R}^n$ (e.g. MNIST in $\mathbb{R}^{784}$, $n = 784 = 28 \times 28$ pixels)
- The data domain is mostly composed of meaningless noise: data occupy a thin region of it!

Main result:

1. A partially trained neural network decomposes the data domain in $\mathbb{R}^n$ as the disjoint union of submanifolds (the **leaves** of a foliation).
2. The dimension d of every submanifold (every leaf of the foliation) is bounded by the number of classes C of our classification model: $d << n$ (e.g. MNIST $d = 9 << 784$).

Data domain and noise

The data domain is the disjoint union of subdomains (foliation).



Data domain as foliation

Main Result/1. Let w be the weights of a deep ReLU neural network classifier, p given by softmax, $G(x, w)$ the data information matrix. The distribution in an open set of the data domain:

$$x \mapsto \mathcal{D}_x = (\ker G(x, w))^{\perp}$$

is involutive i.e.

$$[X, Y] \in \mathcal{D}, \qquad \forall\, X, Y \in \mathcal{D}.$$

Main result/2.

1. At each point in the dataset in $\mathbb{R}^n$, $\ker G(x, w)^{\perp}$ is tangent to a submanifold (**data leaf**) of dimension $\mathrm{rank}\ G(x, w) < C$
2. G defines a foliation on $\mathbb{R}^n$ of rank at most $C - 1$ (**Frobenius Thm**).

Remark: This is not true for the distribution via the Fisher matrix!

$$w \mapsto \mathcal{D}'_w := (\ker F(w))^{\perp}$$

is **not** involutive (e.g. MNIST, lenet).

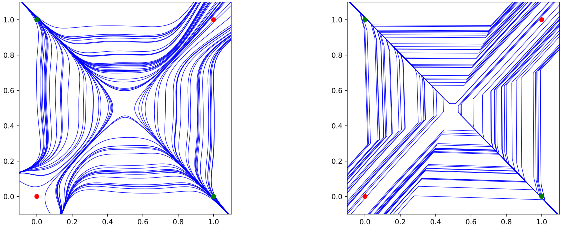
Riemannian Structure on the Data domain

Facts

- The matrix $G(x, w)$, restricted to the subspace $(\ker G(x, w))^{\perp}$ gives a **sub Riemannian** metric to each leaf of the foliation.
  Its rank is not constant *even when restricted to a leaf!*
  (singular foliation theory)
- For a ReLU CNN, the distribution $\mathcal{D}$ defined by the data information matrix $G(x, w)$ is NOT smooth (smooth only on an open set).
- Data leaf: a leaf of the foliation containing some data points.
  We perform dimensionality reduction!
- Extra difficulty: data is contained in a cube (manifold with border and corners!)

Foliation Structure on the Data domain

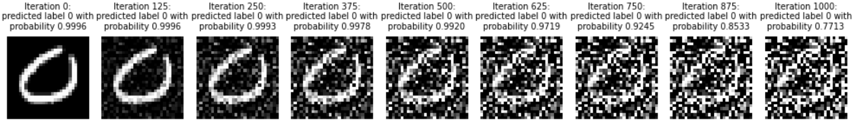
GeLU (left): gives a smooth but not involutive distribution.
ReLU (right): gives a non smooth but involutive distribution.

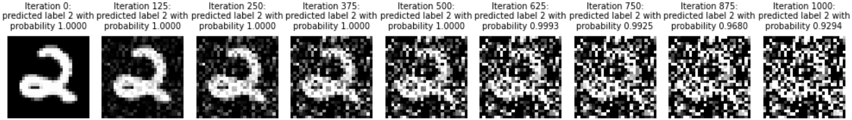


Non linearity	dim $\mathcal{D}_x$	dim span $\{\mathcal{D}_x, [\mathcal{D}_x, \mathcal{D}_x]\}$
ReLU	9	9
GeLU	9	44.84
Sigmoid	9	45

Applications: Denoising, Adversarial Attacks

When moving **away** from a given data leaf, noise is added, but the accuracy remain high.



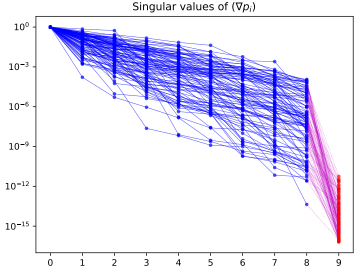


Experiments performed on MNIST with Lenet architecture.

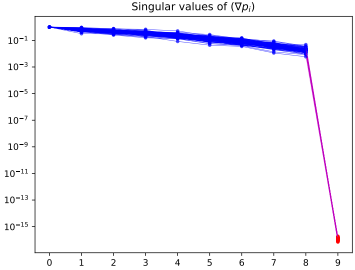
Applications: Knowledge Transfer/1

Eigenvalues for the Data Information Matrix (MNIST dataset)

Data Points

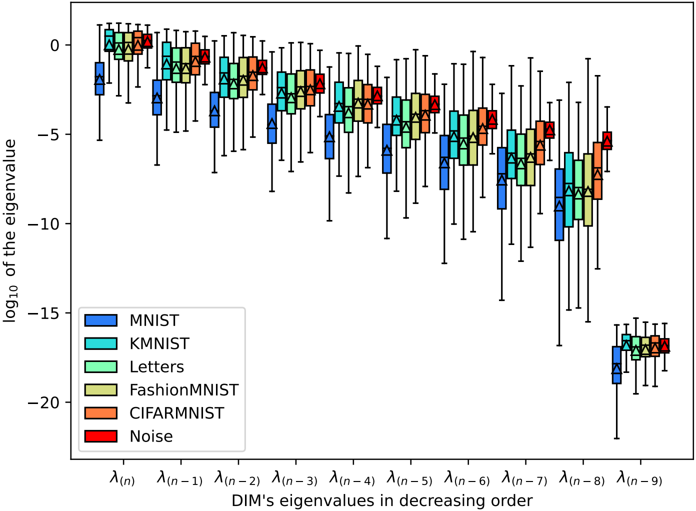


Random Points



Applications: Knowledge Transfer/2

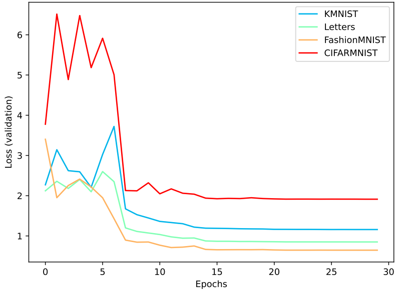
Measuring "distance" between datasets



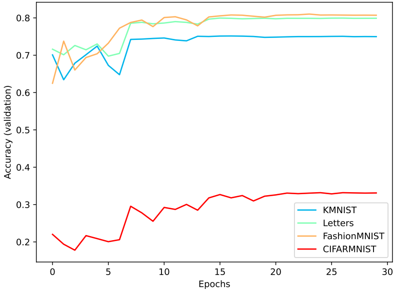
Applications: Knowledge Transfer/3

Measuring "training distance" between datasets

Loss



Accuracy



Dataset	Highest evalue	Lowest evalue	Δ	DIM Trace	Val. Acc.
MNIST	-1.78	-8.58	6.70	-1.52	98%
KMNIST	0.49	-7.75	7.76	0.37	75%
Letters	0.11	-7.99	7.82	0.48	80%
Fashion-MNIST	0.14	-8.08	7.76	0.12	81%
CIFARMNIST	0.41	-6.90	6.75	0.27	33%
Noise	0.24	-5.36	5.49	0.27	NA

Conclusions

- Using a partially trained model we can construct low dimensional submanifolds the **data leaves** of $\mathbb{R}^n$ related with the data the model was trained with.
- We can navigate the data leaves and obtain either data or points with similarities to our data.
- Moving orthogonally to the data leaves will add noise to data, but the model will not change its accuracy.
- Applications:
  - Denoising of images: Project a noisy data point on the data leaves to perform denoising.
  - Knowledge transfer: Use the datamatrix to define the distance between datasets.

Future Directions

We need to understand the geometry and the metric structure of the data leaves.

- It not a riemannian and not a subriemannian manifold: **protosubriemannian geometry, Lie algebroids** language.
- The involutive distribution defining the data leaves is not constant rank: we have a singular foliation!
- What are the geodesics in this geometry? (proto-sub riemannian geometry)
- Navigating the data leaves can lead to data augmentation and efficient denoising algorithms.
- Measuring dataset distance for effective Knowledge Transfer.

Bibliography

- Amari, S.-I. *Natural gradient works efficiently in learning. Neural computation*, 10(2):251–276, 1998.
- Grementieri, L., Fioresi, R. *Model-centric Data Manifold: the Data Through the Eyes of the Model*, SIAM Journal on Imaging Sciences, pp. 1140 – 1156, 2022.
- R. Fioresi, F. Zanchetta *Deep Learning and Geometric Deep Learning: an introduction for mathematicians and physicists*, Int. Jour. Geom. Meth. in Mod. Phys. 2023.
- E. Tron, R. Fioresi, *Manifold Learning via Foliations and Knowledge Transfer*, preprint 2024.
- Sommer, S. and Bronstein, A. M. *Horizontal flows and manifold stochastics in geometric deep learning*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

5. Thermodynamic inspired parameter pruning in DL and GDL

Thermodynamics and SGD

The SGD update of the weights of a (geometric) deep learning model:

$$\mathbf{w} \to \mathbf{w} - \eta \nabla_{\mathcal{B}} L(\mathbf{w}) \qquad \nabla_{\mathcal{B}} L := \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla L_i$$

η: learning rate.
Stochastic differential equation (Ito formalism):

$$dw(t) = -\eta \nabla L(w) dt + \sqrt{2\zeta^{-1} D(w)} dW(t) \tag{2}$$

$W(t)$ models the stochasticity of the SGD
$D(w)$ *diffusion matrix* controls the anisotropy
$\zeta = \eta/(2|\mathcal{B}|)$ *temperature* captures the amount of noise due to SGD.

Reference. Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. 2018 ICLR.

Temperature of Filters of a Neural Network

$$\mathcal{T}(t) = \frac{\mathcal{K}(t)}{k_B\, d} = \frac{1}{k_B\, d} \sum_{k=1}^{d} \frac{1}{2} m_k\, v_k(t)^2 \tag{3}$$

where $v_k(t)$ is the instantaneous velocity of the parameter w_k:

$$v_k(t) = \frac{w_k(t) - w_k(t-1)}{\Delta t} \tag{4}$$

m_k is the mass of parameter w_k and it is set to 1.
The *thermodynamic temperature* is then the time average of $\mathcal{T}(t)$:

$$T = \frac{1}{\tau} \int_0^{\tau} \mathcal{T}(t)\, dt = \frac{1}{\tau\, k_B\, d} \int_0^{\tau} \mathcal{K}(t) \tag{5}$$

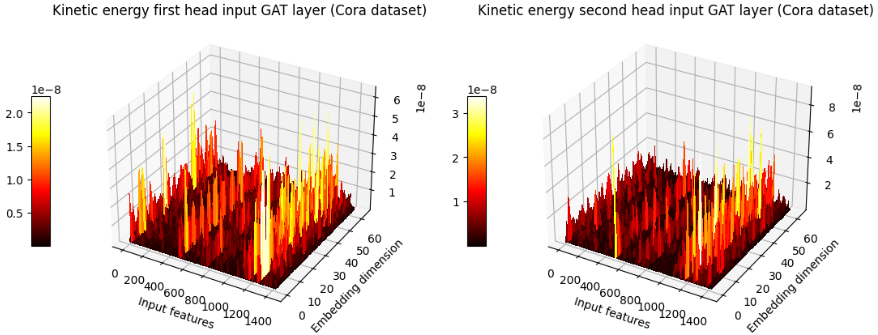
Pruning Hot and Cold Filters in Deep Learning

Model: Lenet
Dataset: MNIST

Model	Test Accuracy	Test Loss
Original model	98.80 ± 0.13 %	0.084 ± 0.022
Without the two "hottest" filters	98.52 ± 0.32 %	0.94 ± 0.36
With only the three "hottest" filters	19.60 ± 5.66 %	5.23 ± 1.61
Without the two "coldest" filters	65.40 ± 13.77 %	2.83 ± 2.22
With only the three "coldest" filters	88.88 ± 6.86 %	0.62 ± 0.48

Table: Accuracy and loss on the test set after cropping different filters from the first CNN on MNIST, in absence of regularization.

Weights and Features in Geometric Deep Learning



Pruning Hot and Cold Features in Geometric Deep Learning

Pruning ratio (%)	"cold" features	"hot" features
0	95.18 ± 0.61 %	95.18 ± 0.61 %
7	95.14 ± 0.35 %	84.36 ± 1.14 %
14	95.12 ± 0.36 %	78.17 ± 1.81 %
28	95.11 ± 0.64 %	67.12 ± 2.95 %
35	95.07 ± 0.40 %	63.28 ± 2.12 %
42	95.00 ± 0.51 %	61.08 ± 2.08 %
63	94.66 ± 0.71 %	55.70 ± 1.92 %
70	94.44 ± 0.60 %	55.00 ± 1.00 %
88	92.43 ± 0.60 %	51.31 ± 1.63 %
95	86.03 ± 1.00 %	50.46 ± 1.60 %

References

- Rita Fioresi, Francesco Faglioni, Francesco Morri, and Lorenzo Squadrani. On the thermodynamic interpretation of deep learning systems. Geometric Science of Information: 5th International Conference, 2021.
- M. Lapenna, F. Faglioni and R. Fioresi Thermodynamics Modeling of Deep Learning Systems, Frontiers in Physics, 2023.
- M. Lapenna, R. Fioresi, F. Faglioni, G. Bruno, Graph Neural Networks and a temperature based pruning technique, preprint 2024.
- Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. International Conference on Learning Representations ICLR 2018.