Decorrelating neurons using persistence

Rubén Ballester RUBEN.BALLESTER@UB.EDU
Carles Casacuberta CARLES.CASACUBERTA@UB.EDU
Sergio Escalera SESCALERA@UB.EDU
Departament de Matemàtiques i Informàtica
Universitat de Barcelona
Gran Via de les Corts Catalanes, 585, 08007 Barcelona, Catalonia, Spain

Editors: Sophia Sanborn, Christian Shewmake, Simone Azeglio, Nina Miolane.

Abstract

We propose a novel way to regularise deep learning models by reducing high correlations between neurons. For this, we present two regularisation terms computed from the weights of a minimum spanning tree of the clique whose vertices are the neurons of a given network (or a sample of those), where weights on edges are correlation dissimilarities. We explore their efficacy by performing a set of proof-of-concept experiments, for which our new regularisation terms outperform some popular ones. We demonstrate that, in these experiments, naive minimisation of all correlations between neurons obtains lower accuracies than our regularisation terms. This suggests that redundancies play a significant role in artificial neural networks, as evidenced by some studies in neuroscience for real networks. We include a proof of differentiability of our regularisers, thus developing the first effective topological persistence-based regularisation terms that consider the whole set of neurons and that can be applied to a feedforward architecture in any deep learning task such as classification, data generation, or regression.

Keywords: Topological data analysis, persistent homology, optimisation, regularisation, neuron correlation, deep learning

1. Introduction

Neural networks have proven to be powerful models to solve complex tasks. Usual neural networks show a high capacity to generalise properly beyond the training dataset used to fit their parameters (Zhang et al., 2021). Although there is no general explanation of why this happens yet, abundant literature is available to tackle this problem (Dziugaite et al., 2020; Graf et al., 2022; Jiang et al., 2020; Jiang* et al., 2020; Kawaguchi et al., 2022). Moreover, many regularisation methods have been proposed to improve generalisation capacity from both theoretical and practical perspectives. According to experimental results, explicit regularisation may improve generalisation performance (Zhang et al., 2021).

Evidence from neuroscience indicates that correlation among human neurons is a significant factor in the brain's ability to encode and process information (Cohen and Kohn, 2011; Kohn and Smith, 2005). This was pointed out in Jin et al. (2020), where it was observed that the generalisation error of a deep network is monotonic with respect to the correlation between weight matrices of neurons or filters, suggesting that decreasing this correlation can be beneficial to improve the generalisation capacity of a network. Furthermore, in Cogswell et al. (2016), overfitting of neural networks was reduced by decorrelating

their neuron activations. Overall, these studies suggest that a reduced correlation between neuron activations could improve the robustness of a network.

However, neuroscience also suggests that redundancy appears naturally in brain circuits and is useful to perform certain computations (Hennig et al., 2018; Mizusaki and O'Donnell, 2021). For this reason, aggressively minimising correlations between all activations or weights may be detrimental for the performance of a neural network.

In this work, we propose a way to minimise only the most relevant high correlations between neurons. For each batch of data during training, we compute regularisation terms based on edge weights of a minimum spanning tree of the clique generated by the most relevant neurons for the batch, based on an importance measure inspired by the activation criterion for neural network pruning presented in Molchanov et al. (2017). Edge weights in the clique are pairwise correlations between activation vectors of neurons. In order to prove that our regularisation terms are almost everywhere differentiable, we use the differential calculus framework for persistent homology developed in Leygonie et al. (2022).

Determining edge weights of a minimum spanning tree is an efficient way of computing zero-dimensional persistent homology from a matrix of correlation dissimilarities. The usefulness of persistent homology to build regularisers is justified by results in Ballester et al. (2022), where it was shown that topological summaries associated with lower activation correlations can be associated with an increased generalisation capacity of the network.

1.1. Contributions

The main contributions of our work can be described as follows:

1. We propose two novel regularisation terms that minimise only some of the highest correlations of the most relevant neurons in a specific training batch. This approach allows for some redundancy in the neural network. Each regularisation term employs a distinct method, and each of them outperforms the other in specific networks.

2. We use differentiable persistence descriptors to ensure differentiability of our regularisation terms, thus developing, to the best of our knowledge, the first topological regularisation terms that depend on the whole set of hidden internal representations of the neurons of a neural network.

3. We provide a set of proof-of-concept experiments to validate the effectiveness of our topological regularisation terms, and find that our regularisers achieve a better performance than several popular regularisation terms in these experiments.

2. Related work

Many works in deep learning study explicit regularisation to improve the generalisation capacity of neural network models. Popular approaches include dropout (Srivastava et al., 2014), in which neurons are dropped randomly during training, and the classical l_1 and l_2 regularisation terms (Tibshirani, 2011), that control the size of weights of a neural network. Among many existing regularisation approaches, some have used correlations between weights and activations of neurons for regularisation. In particular, Jin et al. (2020) uses weight correlations for convolutional and fully-connected layers, and Cogswell et al. (2016)

proposes to minimise a loss function computed from a covariance matrix of neuron activations on a batch. However, both methods have limitations: In Jin et al. (2020), the definition of weight correlation is hardcoded only for fully-connected and convolutional layers, while in Cogswell et al. (2016), correlations between neurons are computed for a set of neurons defined by the user, which is not an easy task for large networks. Both articles reduce all correlations at the same time, without taking into consideration that redundancy between neurons can be important. While the regularisation terms proposed in these articles use only layerwise correlations, without taking into account interactions between neurons of different layers, we propose two regularisation terms that can be used in any feedforward architecture and are not restricted to correlations in the same layer, and moreover they only decorrelate the neurons with highest correlations, thus allowing the network to remain flexible enough to keep an amount of redundancy that could be useful for the task.

During the last years, the popularity of topological methods in machine learning has rapidly increased. An overall survey of these methods can be found in Hensel et al. (2021). In Carrière et al. (2021); Leygonie et al. (2022), frameworks for differential calculus on persistence barcodes were defined, allowing to optimise point cloud shapes and thus to construct topological regularisation terms. In Chen et al. (2019a), the first regularisation term for neural networks based on the topology of the decision region was proposed. From there, specific topological regularisation terms have been discussed for image segmentation (Byrne et al., 2021; Clough et al., 2022; Hu et al., 2019, 2021), autoencoder latent space (Hofer et al., 2019), and classification using decision boundaries (Chen et al., 2019b).

Among the current approaches on regularising neural networks, the most similar to our method are the ones suggested in Birdal et al. (2021); Hofer et al. (2020). In Hofer et al. (2020), zero-dimensional persistent homology is used to optimise the mass concentration of the internal representations in the last hidden layer assuming that the mini-batches used during training are equally distributed among all classes. In Birdal et al. (2021), an upper bound of the generalisation gap of a neural network $\mathcal{N}$ based on persistent homology of the set of weights generated during the training of $\mathcal{N}$ was found and minimised.

Although the existing topological regularisation approaches perform satisfactorily, most of them are restricted to specific tasks or have limitations that can be complemented with our approach, which is fundamentally different from previous methods for the following reasons: 1. Topological regularisation terms based on decision boundaries or the evolution of the whole set of weights do not consider the whole structure of a network, that provides substantial information about generalisation (Corneanu et al., 2020, 2019; Kawaguchi et al., 2022; Rieck et al., 2019; Ballester et al., 2022). 2. Topological regularisation terms designed for specific tasks are too restrictive in general scenarios. Our topological regularisation terms based on correlations of neurons are agnostic to the problem (regression, classification, generative models, etc.) and compatible with most model types, and they work *on the whole structure* of the network by means of neuron activations.

3. Methodology

The presence of high correlations between neurons in a neural network may imply the existence of redundant features learned from the data. This fact suggests that excessively high correlations between neurons restrict the network's capacity to fully utilise its expressivity.

However, entirely avoiding correlated features may be detrimental for learning tasks. This is because (1) it imposes hard restrictions to the weights of neural networks during training, and (2) there is evidence from neuroscience suggesting that correlation is beneficial in brain operation (Hennig et al., 2018; Mizusaki and O'Donnell, 2021). In this paper, we propose to generate a balanced amount of correlation between neurons by reducing only some of the highest ones. To do this, we use zero-dimensional persistent homology to build two regularisation terms that work in a complementary way.

For a thorough introduction to persistent homology, we refer the reader to Edelsbrunner and Harer (2022). To a finite set X and a symmetric function $d\colon X \times X \to \mathbb{R}_{\geq 0}$ such that $d(x,x) = 0$ for all $x \in X$ one associates a *persistence diagram* in every homological dimension greater than or equal to zero (details are given in Appendix A). In this work we only use persistent homology in dimension zero. Since points in a zero-dimensional persistence diagram are aligned along the positive y axis, we only focus on their y-coordinates. Hence we associate to each pair (X,d) as above a finite multiset $D(X,d)$ of positive real numbers, ignoring points at infinity. Such numbers are the nonzero weights of the edges of a minimum spanning tree (MST) of the undirected weighted complete graph (V,E,w) with vertices $V = X$ and weights $w(\{u,v\}) = d(u,v)$. A proof of this fact is given in Appendix A.

Let $\mathcal{N}\colon \mathcal{X} \to \mathcal{Y}$ denote a feedforward neural network, and let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subseteq \mathcal{X} \times \mathcal{Y}$ be a dataset —in our case, batches of the training dataset. Let $G_{\mathcal{N}}$ be the complete graph of the network $\mathcal{N}$, whose vertices are the neurons of $\mathcal{N}$. As we do not know the marginal distribution of the data on $\mathcal{X}$, we approximate the correlation between two neurons seen as random variables $u, v\colon \mathcal{X} \to \mathbb{R}$ using the sample correlation for the neuron activations in the dataset $\mathcal{D}$. The *sample correlation* of two vectors $x, y \in \mathbb{R}^n$ is defined as

$$\operatorname{corr}(x,y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}},$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$. The correlation between two neurons u, v is $\operatorname{corr}_{\mathcal{D}}(u,v) \triangleq \operatorname{corr}\left(u(\mathcal{D}), v(\mathcal{D})\right)$ where $u(\mathcal{D}) \triangleq (u(x_1), \ldots, u(x_n))$, $v(\mathcal{D}) \triangleq (v(x_1), \ldots, v(x_n))$.

In our case, neurons are considered to be similar when they share a large correlation in absolute value. Since correlations take values between -1 and 1, we define the *correlation dissimilarity* between neurons as a function $d\colon V(G_{\mathcal{N}}) \times V(G_{\mathcal{N}}) \to [0,1]$ given by $d(u,v) = 1 - |\operatorname{corr}_{\mathcal{D}}(u,v)|$. Given this dissimilarity function, we can study correlations between any subset of neurons $V' \subseteq V(G_{\mathcal{N}})$ by means of the persistence diagram $D(V',d)$.

By the cut property of minimum spanning trees, each MST of a graph contains, for each of its vertices, at least one edge with the minimum weight among its incident edges. In our case, as we use complete graphs, this is translated into the fact that the diagram $D(V',d)$ contains, for each neuron $u \in V'$, an incident edge to u achieving the value

$$\min_{v \in V',\, u \neq v} \left(1 - |\operatorname{corr}_{\mathcal{D}}(u,v)|\right) = 1 - \max_{v \in V',\, u \neq v} |\operatorname{corr}_{\mathcal{D}}(u,v)|$$

among, possibly, other high correlations to form the MST. Therefore, by maximising the values of $D(V',d)$, we are in fact minimising a set of correlations between neurons in V' containing the highest correlations achieved by neurons in the set.

Current neural networks contain an enormous quantity of neurons and computing a MST of the complete graph $(V(G_{\mathcal{N}}), d)$ is not feasible in many cases, as computing a MST has a

complexity of $\mathcal{O}(e \cdot \alpha(e, v))$ (Chazelle, 2000, Theorem 1.1), where $\alpha(e, v)$ is the functional inverse of Ackermann's function Tarjan (1975) and e and v are the number of edges and vertices, respectively, with $e = \binom{v}{2}$ because the graph is a clique.

For this reason, we consider, for each batch $\mathcal{B} = \{(x_1, y_1), \ldots, (x_{|\mathcal{B}|}, y_{|\mathcal{B}|})\}$ during training, a subset of neurons $V_\mathcal{B} \subseteq V(G_\mathcal{N})$ that may have smaller cardinality than $V(G_\mathcal{N})$. In particular, for the cases in which we cannot set $V_\mathcal{B} = V(G_\mathcal{N})$, we sample $V_\mathcal{B}$ using an *importance sampling algorithm* for each batch. We take the top percentage P of most important neurons of each layer given the batch, except for the last layer, where we take all the neurons, where P is a hyperparameter depending on the size of the neural network. This is because the last layer showed to contain relevant information with respect to generalisation in other works like the one by Carlsson and Gabrielsson (2020). For our experiments with very large neural networks, we set P to 0.5% due to practical hardware limitations.

The *importance* of a neuron given a batch is set to the average quantity of absolute activation achieved by the neuron, and it is inspired by the activation criterion for pruning presented in Molchanov et al. (2017, Section 2.2). The higher this value for a neuron is, the more relevance we allot to the neuron. More precisely, denote $\bar{v}_\mathcal{B} = |\mathcal{B}|^{-1} \sum_{i=1}^{|\mathcal{B}|} |v(x_i)|$ and let $V(G_\mathcal{N})_l = \{v_i\}_{i=1}^{n_l}$ be the neurons of the l-layer of $\mathcal{N}$ in any descending order of their $\bar{v}_\mathcal{B}$ values. In our case, the order is given by the `argsort` function of TensorFlow. We use

$$V_\mathcal{B} = \bigcup_l V_{\mathcal{B},l} \quad \text{where } V_{\mathcal{B},l} = \begin{cases} \{v_1, \ldots, v_{\lfloor 0.005\, n_l \rfloor}\} \text{ if } l \neq L, \\ V(G_\mathcal{N})_l \text{ otherwise,} \end{cases}$$

where l iterates over all possible layers of $\mathcal{N}$ and L is the number of the last layer.

Since regularisation terms are minimised by network training algorithms, to maximise a function $f(\theta)$ we minimise its opposite function $-f(\theta)$. We propose two regularisation terms that maximise persistence diagram values in different ways:

$$\mathcal{T}_1(\theta) \triangleq -\sum_{y \in D(V_\mathcal{B}, d)} y, \qquad (1) \qquad \mathcal{T}_2(\theta) \triangleq -\alpha \bar{D}(V_\mathcal{B}, d) + \beta \sigma(D(V_\mathcal{B}, d)), \qquad (2)$$

where $\alpha, \beta \in \mathbb{R}_{\geq 0}$ are weight parameters, θ is the set of parameters of the network, and

$$\bar{D}(V_\mathcal{B}, d) = \frac{1}{|D(V_\mathcal{B}, d)|} \sum_{y \in D(V_\mathcal{B}, d)} y, \qquad \sigma^2(D(V_\mathcal{B}, d)) = \frac{1}{|D(V_\mathcal{B}, d)|} \sum_{y \in D(V_\mathcal{B}, d)} (y - \bar{D}(V_\mathcal{B}, d))^2$$

are the mean and variance of the values of $D(V_\mathcal{B}, d)$. The regularisation term $\mathcal{T}_1$ given by Equation (1) maximises the sum of values in the persistence diagram, while the regularisation term $\mathcal{T}_2$ given by Equation (2) focuses on how the entries of the persistence diagram are distributed, minimising their dispersion and maximising their average value. In our case, we pick $\alpha = \beta = 1/2$ since we treat mean and dispersion with the same strength.

Theorem 1 *Let $c, n \geq 2$ and let $d\colon \mathbb{R}^n \times \mathbb{R}^n \to [0, 1]$ be $d(x, y) = 1 - |\mathrm{corr}(x, y)|$ where* corr *denotes correlation. There is an open dense subset $\mathfrak{D}_{c,n} \subseteq \mathbb{R}^{cn}$ such that the functions*

$$\mathcal{T}_1(x_1, \ldots, x_c) \triangleq -\sum_{y \in D(X, d)} y, \qquad \mathcal{T}_2(x_1, \ldots, x_c) \triangleq -\alpha \bar{D}(X, d) + \beta \sigma(D(X, d))$$

are $\mathcal{C}^\infty$ on $\mathfrak{D}_{c,n}$ for all $\alpha, \beta \in \mathbb{R}_{\geq 0}$, where $X = \{x_1, \ldots, x_c\}$ and $\bar{D}(X, d)$ and $\sigma(D(X, d))$ denote average and standard deviation of the zero-dimensional persistence diagram $D(X, d)$.

A proof of this result is provided in Appendix B. By the chain rule, our regularisation terms are well defined as soon as the neuron activations of the set of neurons $V_{\mathcal{B}} = \{\nu_1, \dots, \nu_c\}$ in the batch $\mathcal{B} = \{x_1, \dots, x_n\}$ form a vector

$$\mathbf{v} = (\nu_1(x_1), \dots, \nu_1(x_n)), \dots, (\nu_c(x_1), \dots, \nu_c(x_n)) \in \mathbb{R}^{cn}$$

such that $\mathbf{v} \in \mathfrak{D}_{c,n}$ and such that the neuron activations are obtained in a differentiable way. Experimentally, we need not control when this vector $\mathbf{v}$ is inside $\mathfrak{D}_{c,n}$ thanks to the fact that $\mathfrak{D}_{c,n}$ is a dense set. However, we note that ignoring points where non-differentiability may occur in the domain could introduce errors in some iterations during training, as it may also happen in fact with ReLU (Bertoin et al., 2021).

4. Results

In this section, we first describe the experimental setup and the computational resources that we use to validate the hypothesis stated in the previous section. This is done in Subsection 4.1. Then, we present and discuss the results in Subsection 4.2. The code used to perform these experiments is linked as supplementary material[1].

4.1. Experimental setup

In this section, we present two blocks of basic proof-of-concept experiments to demonstrate the plausibility of the hypotheses formulated in Section 3. For each block, we train several neural networks following a common architecture with different regularisation terms, including our proposed ones, and without regularisation terms. For the first block, we use multilayer perceptron models whereas we use VGG-like models for the second one. The networks of the first block are trained in the MNIST dataset whereas the networks of the second one are trained in CIFAR-10. In both blocks, we explore the same set of weights for the regularisation terms. Finally, to compare the accuracies of our proposed regularisation terms to the other alternatives, we use the Friedman statistical test with its Nemenyi post-hoc. Further details of the experiments are provided through this section.

Multilayer perceptron experiments. In the first block of experiments, we examine our regularisation terms in a simplified problem. We train three different multilayer perceptron architectures with 1000 hidden neurons, labelled `0`, `1`, and `2` using the MNIST dataset (LeCun et al., 2010). Networks `0` and `1` share the same fully connected architecture. However, network `1` is trained using dropout with a 50% probability of dropping a hidden neuron at each iteration. Specifically, architectures `0` and `2` have a trapezium shape consisting of a sequence of hidden layers of 450, 350, and 200 neurons for the first network, and of 300, 250, 200, 150, and 100 for the second one, respectively. In this block of experiments, we do not aim to achieve state-of-the-art performance but to test our approach in a simple scenario where no sampling of neurons is needed to compute persistence diagrams.

PGDL experiments. In the second block of experiments, the objective is to see if the method scales well to more complex datasets and models. We train eight different neural network architectures from the PGDL dataset introduced in Jiang et al. (2020) during the

1. https://github.com/rballeba/DecorrelatingNeuronsUsingPersistence

NeurIPS 2020 competition track. The PGDL dataset is a collection of tasks where each task is composed of a set of different neural network architectures with different generalisation capabilities trained with a common dataset. The eight different neural network architectures we take belong to the first task, which is composed of VGG-like neural networks trained in the CIFAR10 dataset Krizhevsky et al. (2009). The architectures we selected are the ones that correspond to the numbers `20`, `21`, `22`, `23`, `148`, `149`, `150`, and `151` from the dataset. Architectures `20`, `21`, `148`, and `149` are the same as the architectures `22`, `23`, `150`, and `151`, but with a layerwise dropout probability of 0.5, respectively. The difference between models `22` and `23` is the width of their convolutions, where architecture `22` has convolution widths of 256 and architecture `23` has convolution widths of 512. Finally, the architectures `150` and `151` are the same as the architectures `22` and `23` but with one more dense layer.

Training procedures. In these experiments, we train different architectures with regularisation terms weighted with several values. To train the networks, we replicate approximately the training performed by the PGDL dataset used in the second block of experiments.

For both blocks of experiments we split the data into training, validation, and test datasets. For the MNIST dataset, we split the original training dataset into new training and validation datasets with 80% and 20% of the original data, respectively. Finally, we use the original test dataset for testing. For the CIFAR10 dataset, we split the original training dataset into new training and validation datasets, where we choose 1000 examples of each class randomly for the validation dataset and we place the remaining examples into the training dataset. Again, we reuse the original test dataset.

For the training procedure, we train for a maximum of 1200 epochs with early stopping after 20 epochs without improvement in accuracy and with a batch size of 256. The algorithm used for training is the usual stochastic gradient descent (SGD) with momentum 0.9. For the first block of experiments, we use an adaptive learning rate $\alpha_i \triangleq \alpha_0 \cdot (0.95)^{i/3520}$ where i is the iteration where the learning rate α_i is used and $\alpha_0 = 0.01$. For the second block of experiments, we use a fixed learning rate of 0.001, for which we obtained similar accuracies to the ones given by the original trainings of the PGDL neural networks.

Let $\mathrm{CCE}(\theta)$ denote the categorical cross entropy loss for a fixed neural network, clear from the context, with a set of parameters θ. For each of the networks described before we perform several trainings with the different regularisation terms that we study, weighed by different values. In particular, each training minimises a loss function

$$\mathcal{L}(\theta) = \mathrm{CCE}(\theta) + \omega\mathcal{R}(\theta), \tag{3}$$

where $\omega \in \{10^{-6}, 10^{-5}, 10^{-4}, 0.001, 0.01, 0.1, 1.5, 10, 100\}$ represents one of the possible weight values used in the experiments, and $\mathcal{R}(\theta)$ represents one of the regularisation terms. We also train the models without any regularisation term, i.e., with $\omega = 0$.

We use the full and sampled version of our regularisers $\mathcal{T}_1$ and $\mathcal{T}_2$ for the first and second blocks of experiments, respectively, due to the small size of the networks of the first block and to the large size of the networks of the second one. To see if reducing only some correlations between neurons is better than minimising all of them, we also study the regularisation term

$$\mathcal{C}(\theta) = \frac{1}{|\mathfrak{C}|} \sum_{(x,y)\in\mathfrak{C}} |\mathrm{corr}_{\mathcal{B}}(x, y)|, \tag{4}$$

Table 1: **Test accuracies for different training procedures and networks.** Each row represents training with a regularisation term except for the first row, that represents no regularisation. $\mathcal{T}_1$: First topological regularisation term (1); $\mathcal{T}_2$: Second topological regularisation term (2); l_1: Lasso regression regularisation term; l_2: Ridge regression regularisation term; $\mathcal{C}$: Regularisation term minimising all pairwise correlations in $V_{\mathcal{B}}$ (4); $\emptyset$: no regulariser. Each column corresponds to a network in the PGDL dataset. Best accuracies per network are bolded.

	MNIST and MLP			PGDL and VGG-like							
	0	1	2	20	21	22	23	148	149	150	151
$\emptyset$	**0.929**	0.501	0.636	0.681	0.680	0.685	0.682	0.672	0.677	0.675	0.680
$\mathcal{T}_1$	0.928	**0.547**	**0.883**	0.687	**0.705**	0.675	0.700	0.688	**0.704**	0.678	**0.698**
$\mathcal{T}_2$	0.923	0.540	0.879	**0.691**	0.701	**0.688**	**0.706**	**0.689**	0.698	**0.688**	0.695
l_1	0.914	0.536	0.870	0.682	0.680	0.682	0.683	0.677	0.675	0.685	0.678
l_2	0.919	0.531	0.878	0.681	0.688	0.686	0.683	0.680	0.680	0.681	0.679
$\mathcal{C}$	0.923	0.530	0.881	0.679	0.687	0.680	0.686	0.678	0.690	0.683	0.674

where $\mathfrak{C} = \{(x, y) \in V_{\mathcal{B}} \times V_{\mathcal{B}} : x \neq y \text{ and } \text{corr}_{\mathcal{B}}(x, y) \neq 0\}$, and $V_{\mathcal{B}}$ is defined as in Section 3. Note that for the first block of experiments we consider all the non-input neurons and for the second block of experiments we perform the same sample of neurons due to the complexity of computing all the possible pairwise correlations for each iteration of the training. Finally, we also train the networks with l_1 and l_2 regularisation terms (Tibshirani, 2011).

Evaluation procedure. To evaluate the performances of the regularisation terms compared, we use a Friedman test with the Nemenyi post-hoc, as proposed in Demšar (2006), and we report the test accuracies for each regularisation term and network. To obtain test accuracies, we choose, for each term and network, the weight that maximises the validation accuracy after training. Then, we compute the test accuracy using the selected weight. For the training procedures without regularisation terms, we compute test accuracies directly.

4.2. Results and analysis

Table 1 contains test accuracies for all the networks and regularisation terms studied in our experiments. Each cell shows the test accuracy of the regularisation term with the weight that obtained the best validation accuracy in its column. The Friedman test with null hypothesis that all the algorithms are equivalent (Demšar, 2006) gives a p-value of 0.00001, so we reject the null hypothesis. Therefore, we perform a Nemenyi post-hoc test, obtaining the p-value matrix shown in Table 2. The null hypothesis of this test is that there is no difference between the accuracies yielded by the two training approaches. A critical difference diagram for the Friedman and Nemeny statistical tests is shown in Appendix D.

When comparing test accuracies individually, $\mathcal{T}_1$ and $\mathcal{T}_2$ outperform the other training methods for all the networks except for model `0`. In fact, $\mathcal{T}_1$ and $\mathcal{T}_2$ obtain the best test accuracies in five of the networks each, respectively, out of eleven total networks. Both of

Table 2: **Nemenyi *p*-value matrix.** Cells contain p-values of the Nemenyi post-hoc test (p-values < 0.05 are bolded). The meaning of column labels is specified in Table 1.

	$\emptyset$	$\mathcal{T}_1$	$\mathcal{T}_2$	l_1	l_2	$\mathcal{C}$
$\emptyset$		**0.018**	**0.002**	0.900	0.785	0.900
$\mathcal{T}_1$			0.900	**0.036**	0.380	0.159
$\mathcal{T}_2$				**0.006**	0.122	**0.036**
l_1					0.900	0.900
l_2						0.900

them are significantly better than training without regularisation term, according to the Nemenyi p-value matrix, showing p-values of 0.018 and 0.002 for $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively.

Regarding the differences between minimising all the correlations and minimising only the highest ones, we see that $\mathcal{T}_1$ and $\mathcal{T}_2$ obtain low p-values against $\mathcal{C}$. In particular, for $\mathcal{T}_2$ and $\mathcal{C}$ we obtain a p-value of 0.036, that validates the hypothesis that minimising only the highest correlations is significantly different than minimising all the correlations for a sampled set of relevant neurons in our experiments.

As for classical regularisation terms, both $\mathcal{T}_1$ and $\mathcal{T}_2$ obtain p-values lower than or equal to 0.05 with respect to l_1, making our regularisation terms better than l_1. The p-values of $\mathcal{T}_1$ and $\mathcal{T}_2$ with respect to l_2 are lower than the other p-values for l_2, although not as low as for l_1. This, together with the fact that l_2 never obtains the best accuracies, suggests that our regularisers are better than l_2, although more experiments are needed to confirm this claim. Overall $\mathcal{T}_1$ and $\mathcal{T}_2$ perform better than other training approaches in our experiments.

5. Limitations and future work

The computational cost of determining persistence diagrams can pose limitations in practical scenarios. To extend the applicability of our regularisation terms, improvements to the algorithms for computing zero-dimensional persistence should be developed. For example, advances in the implementation of these algorithms in a distributed manner, as discussed in Rostrup et al. (2013); Sanders and Schimek (2023), could be relevant.

A crucial bottleneck in our pipeline is the computation of pairwise correlations. With a large number of neurons, it is impractical to compute correlations for all neurons at each training step. The effect of selecting a range of different percentages of neurons for sampling in large networks should be tested, since neuron selection may greatly impact the performance of regularisation terms. To enhance the computational efficiency of regularisation focused on high correlations, one approach could involve implementing weighted dropout based on the average correlation of each neuron with others, discarding at each step of the training procedure the nodes with lowest overall average correlation. Selectively training only neurons with higher average correlations may yield a positive effect by forcing them to learn independent functions.

We emphasise that our experiments were performed in simple tasks as a proof of concept of our methodology, and that further work is needed to understand how correlations

affect generalisation capacity, especially in complex scenarios. For example, Birdal et al. (2021) found that their regulariser was more effective in suboptimal hyperparameter settings. A similar phenomenon might occur with our regularisers.

As correlations within neurons are independent of dataset labels, our approach is well-suited for application in unsupervised or semi-supervised learning.

6. Conclusions

In this work, we introduced regularisation terms that minimise high correlations between the most important neurons given a training batch, by maximising the values of their zero-dimensional persistence computed with the dissimilarity function $d(u, v) = 1 - |\text{corr}_{\mathcal{D}}(u, v)|$. The use of persistent homology in Corneanu et al. (2020, 2019) and Ballester et al. (2022) was intended to assess trained models as a post-hoc method. The present article adds evidence that the association between activation correlations and generalisation gap can be exploited to build regularisers with the aim of enhancing generalisation.

Our regularisation terms outperformed classical regularisation terms and improved performance compared to minimising all pairwise correlations of important neurons in the MNIST and CIFAR10 datasets with MLP and VGG-like architectures, respectively. These findings support the hypothesis that neuron correlations play a role in the generalisation capacity of neural networks, consistently with previous studies such as Cogswell et al. (2016); Jin et al. (2020). Additionally, we demonstrated that, when minimising higher correlations using persistent homology, several loss functions that are used with the same objective can yield different performances, suggesting that, for differentiable persistence descriptors, the choice of a loss function is a crucial step in the process.

Our results also show that topological regularisation terms can be used to improve the performance of neural networks not only by considering the final representations of the data, but also by looking at intermediate representations as well. In summary, our findings provide insight into the relevance of topological data analysis and neuron correlations on the generalisation capacity of neural networks, as well as their potential for future advances.

Acknowledgments

This work was supported by the Ministry of Science and Innovation of Spain through the research projects PID2022-136436NB-I00 and PID2020-117971GB-C22; Ministry of Universities of Spain through the contract FPU21/00968 (R. Ballester); Departament de Recerca i Universitats de la Generalitat de Catalunya with reference 2021 SGR 00697 (R. Ballester, C. Casacuberta); and ICREA under the ICREA Academia program (S. Escalera).

References

Rubén Ballester, Xavier Arnal Clemente, Carles Casacuberta, Meysam Madadi, Ciprian Corneanu, and Sergio Escalera. Predicting the generalization gap in neural networks using topological data analysis, 2022. URL https://arxiv.org/abs/2203.12330.

David Bertoin, Jérôme Bolte, Sébastien Gerchinovitz, and Edouard Pauwels. Numerical influence of ReLU'(0) on backpropagation. In M. Ranzato, A. Beygelzimer,

Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 468–479. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/043ab21fc5a1607b381ac3896176dac6-Paper.pdf.

Tolga Birdal, Aaron Lou, Leonidas J Guibas, and Umut Simsekli. Intrinsic dimension, persistent homology and generalization in neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6776–6789. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/35a12c43227f217207d4e06ffefe39d3-Paper.pdf.

Nick Byrne, James R. Clough, Giovanni Montana, and Andrew P. King. A persistent homology-based topological loss function for multi-class CNN segmentation of cardiac MRI. In Esther Puyol Anton, Mihaela Pop, Maxime Sermesant, Víctor Campello, Alain Lalande, Karim Lekadir, Avan Suinesiaputra, Oscar Camara, and Alistair Young, editors, *Statistical Atlases and Computational Models of the Heart. M&Ms and EMIDEC Challenges*, pages 3–13, Cham, 2021. Springer International Publishing. ISBN 978-3-030-68107-4.

Gunnar Carlsson and Rickard Brüel Gabrielsson. Topological approaches to deep learning. In Nils A. Baas, Gunnar E. Carlsson, Gereon Quick, Markus Szymik, and Marius Thaule, editors, *Topological Data Analysis*, pages 119–146, Cham, 2020. Springer International Publishing. ISBN 978-3-030-43408-3.

Mathieu Carrière, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hariprasad Kannan, and Yuhei Umeda. Optimizing persistent homology based functions. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1294–1303. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/carriere21a.html.

Bernard Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *Journal of the ACM*, 47(6):1028–1047, Nov 2000. ISSN 0004-5411. doi: 10.1145/355541.355562. URL https://doi.org/10.1145/355541.355562.

Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2573–2582. PMLR, 16–18 Apr 2019a. URL https://proceedings.mlr.press/v89/chen19g.html.

Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2573–2582. PMLR, 16–18 Apr 2019b. URL https://proceedings.mlr.press/v89/chen19g.html.

J. R. Clough, N. Byrne, I. Oksuz, V. A. Zimmer, J. A. Schnabel, and A. P. King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis; Machine Intelligence*, 44(12):8766–8778, Dec 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2020.3013679.

Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Marlene R. Cohen and Adam Kohn. Measuring and interpreting neuronal correlations. *Nature Neuroscience*, 14(7):811–819, 2011.

Ciprian Corneanu, Meysam Madadi, Sergio Escalera, and Aleix M. Martinez. What does it mean to learn in deep networks? And, how does one detect adversarial attacks? In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4752–4761, 2019. doi: 10.1109/CVPR.2019.00489.

Ciprian Corneanu, Sergio Escalera, and Aleix M. Martinez. Computing the testing error without a testing set. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2674–2682, Los Alamitos, CA, USA, Jun 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.00275. URL https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00275.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006. URL http://jmlr.org/papers/v7/demsar06a.html.

Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M. Roy. In search of robust measures of generalization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Herbert Edelsbrunner and John L. Harer. *Computational Topology: an Introduction*. American Mathematical Society, 2022.

Florian Graf, Sebastian Zeng, Bastian Rieck, Marc Niethammer, and Roland Kwitt. On measuring excess capacity in neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 10164–10178. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/420492060687ca7448398c4c3fa10366-Paper-Conference.pdf.

Jay A. Hennig, Matthew D. Golub, Peter J. Lund, Patrick T. Sadtler, Emily R. Oby, Kristin M. Quick, Stephen I. Ryu, Elizabeth C. Tyler-Kabara, Aaron P. Batista, Byron M. Yu, and Steven M. Chase. Constraints on neural redundancy. *Elife*, 7, August 2018.

Felix Hensel, Michael Moor, and Bastian Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4, 2021. ISSN 2624-8212. doi: 10.3389/frai.2021.681108. URL https://www.frontiersin.org/articles/10.3389/frai.2021.681108.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Mandar Dixit. Connectivity-optimized representation learning via persistent homology. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2751–2760. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hofer19a.html.

Christoph Hofer, Florian Graf, Marc Niethammer, and Roland Kwitt. Topologically densified distributions. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4304–4313. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/hofer20a.html.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2d95666e2649fcfc6e3af75e09f5adb9-Paper.pdf.

Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen. Topology-aware segmentation using discrete Morse theory. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LGgdb4TS4Z.

Yiding Jiang, Pierre Foret, Scott Yak, Daniel M. Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. NeurIPS 2020 competition: Predicting generalization in deep learning, 2020. URL https://arxiv.org/abs/2012.07976.

Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

Gaojie Jin, Xinping Yi, Liang Zhang, Lijun Zhang, Sven Schewe, and Xiaowei Huang. How does weight correlation affect generalisation ability of deep neural networks? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21346–21356. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f48c04ffab49ff0e5d1176244fdfb65c-Paper.pdf.

K. Kawaguchi, Y. Bengio, and L. Kaelbling. Generalization in deep learning. In *Mathematical Aspects of Deep Learning*, pages 112–148. Cambridge University Press, Dec 2022. doi: 10.1017/9781009025096.003. URL https://doi.org/10.1017%2F9781009025096.003.

Adam Kohn and Matthew A. Smith. Stimulus dependence of neuronal correlation in primary visual cortex of the macaque. *Journal of Neuroscience*, 25(14):3661–3673, 2005.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, Technical Report, University of Toronto, Ontario, 2009.

Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. MNIST Handwritten Digit Database, AT & T Labs, 2010. URL http://yann.lecun.com/exdb/mnist.

Jacob Leygonie, Steve Oudot, and Ulrike Tillmann. A framework for differential calculus on persistence barcodes. *Foundations of Computational Mathematics*, 22(4):1069–1131, Aug 2022. ISSN 1615-3383. doi: 10.1007/s10208-021-09522-y. URL https://doi.org/10.1007/s10208-021-09522-y.

Beatriz E. P. Mizusaki and Cian O'Donnell. Neural circuit function redundancy in brain disorders. *Current Opinion in Neurobiology*, 70:74–80, 2021. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2021.07.008. URL https://www.sciencedirect.com/science/article/pii/S0959438821000787.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJGCiw5gl.

Julián Burella Pérez, Sydney Hauke, Umberto Lupo, Matteo Caorsi, and Alberto Dassatti. Giotto-ph: A Python library for high-performance computation of persistent homology of Vietoris–Rips filtrations, 2021.

Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByxkijC5FQ.

Scott Rostrup, Shweta Srivastava, and Kishore Singhal. Fast and memory-efficient minimum spanning tree on the GPU. *International Journal of Computational Science and Engineering (IJCSE)*, 8(1):21–33, Feb 2013. ISSN 1742-7185. doi: 10.1504/IJCSE.2013.052115. URL https://doi.org/10.1504/IJCSE.2013.052115.

Peter Sanders and Matthias Schimek. Engineering massively parallel MST algorithms, 2023. URL https://arxiv.org/abs/2302.12199.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, Apr 1975. ISSN 0004-5411. doi: 10.1145/321879.321884. URL https://doi.org/10.1145/321879.321884.

Robert Tibshirani. Regression shrinkage and selection via the Lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3): 273–282, 2011. doi: https://doi.org/10.1111/j.1467-9868.2011.00771.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2011.00771.x.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, Feb 2021. ISSN 0001-0782. doi: 10.1145/3446776. URL https://doi.org/10.1145/3446776.

Appendix A. Equivalence between zero-dimensional persistence and minimum spanning trees

In this appendix, we prove that there is a bijection between the set of non-diagonal points with finite death parameter in the zero-dimensional persistence diagram of a finite set X equipped with a dissimilarity function $d\colon X \times X \to \mathbb{R}_{\geq 0}$ and the set of weights of a minimum spanning tree of the complete weighted graph whose vertices are the elements of X and whose weights are given by d. We call d a *dissimilarity* if it is symmetric and $d(x,x) = 0$ for all $x \in X$. An example is the Euclidean distance function when X is a set of points in Euclidean space, and another example is the correlation dissimilarity used in this article when X is a set of neurons of a neural network.

The *persistence diagram* $\mathrm{Dgm}_k(V(X,d))$ of $X = \{p_1, \ldots, p_c\}$ in homological dimension k with respect to the given dissimilarity d is an unordered multiset of points $\{(b_i, d_i)\}_{i \in I}$ in $\mathbb{R}^2$ where b_i is the birth parameter and d_i is the death parameter (possibly infinite) of an element in a full set I of linearly independent generators of kth simplicial homology $H_k(V_t(X,d))$ of the *Vietoris–Rips filtered simplicial complex* $V(X,d) = \{V_t(X,d)\}_{t \geq 0}$. The simplicial complex $V_t(X,d)$ at filtration level t has an m-simplex for every collection of points $p_{i_0}, \ldots, p_{i_m}$ in X such that $d(p_{i_r}, p_{i_s}) \leq t$ for $i_r, i_s \in \{1, \ldots, c\}$. By convention, persistence diagrams include all points in the diagonal $\Delta^\infty = \{(x,x) : x \geq 0\}$ with infinite multiplicity. We compute simplicial homology with coefficients in the field of two elements.

For a connected weighted graph $G = (V, E, w)$, we denote the multiset of weights of G by $\mathcal{W}_G = \{w(e) : e \in E\}$. A *minimum spanning tree* of G is a subgraph without cycles containing all the vertices with the minimum possible total edge weight. Kruskal's algorithm (Kruskal, 1956) finds a minimum spanning tree of every weighted graph G and shows that the multiset of weights of all minimum spanning trees of G coincide.

Theorem 2 *Let (X,d) be a finite set and let $d\colon X \times X \to \mathbb{R}_{\geq 0}$ be a symmetric function such that $d(p,p) = 0$ for all $p \in X$. Let G be the complete weighted graph with set of vertices X and weights given by d. Let $D_{<\infty} = \{(b_i, d_i) \in \mathrm{Dgm}_0(V(X,d)) : b_i < d_i < \infty\}$ be the multiset of finite, non-diagonal points of the Vietoris-Rips zero-dimensional persistence*

diagram of (X, d). *Then all points* (b_i, d_i) *in* $D_{<\infty}$ *have* $b_i = 0$ *and*

$$\{d_i : (0, d_i) \in D_{<\infty}\} = \{w \in \mathcal{W}_T : w > 0\}$$

for any minimum spanning tree T *of* G.

Proof Note first that, since the function d takes non-negative values and $d(p, p) = 0$ for all $p \in X$, we have that $V_r(X, d) = \emptyset$ for $r < 0$ and $\{p\} \in V_0(X, d)$ for all $p \in X$. This means that all connected components of $V(X, d)$ are born at $r = 0$ and consequently all $(b_i, d_i) \in \mathrm{Dgm}_0(V(X, d)) \setminus \Delta^\infty$ have $b_i = 0$. The corresponding death values d_i are filtration levels at which connected components merge. Moreover, $V(X, d)$ becomes connected eventually, since X is finite. Hence there is a single point in $\mathrm{Dgm}_0(V(X, d))$ with $d_i = \infty$.

Write $D_{<\infty} = \{(0, d_1), \ldots, (0, d_n)\}$ with $d_1 \leq \cdots \leq d_n$ without loss of generality, counted with the respective multiplicities. For each connected component C_i in $V_0(X, d)$, choose a minimum spanning tree T_i with total weight 0, and write the edges of T_i as $e^i_1, \ldots, e^i_{k_i}$. Next, order the edges of the complete graph G in such a way that the first elements of the list are $e^1_1, \ldots, e^1_{k_1}, \ldots, e^n_1, \ldots, e^n_{k_n}$ in any order.

For each point $(0, d_i)$ in $D_{<\infty}$, there is at least one edge e_i in G with weight d_i connecting two previously separated connected components in the Vietoris-Rips filtration. Place the edges $e_1, \ldots, e_n$ before all the edges in G with their same weights. If there exist $i \neq j$ such that $w(e_i) = w(e_j)$, order them arbitrarily and place them consecutively. Then, by applying Kruskal's algorithm, we obtain a minimum spanning tree T of G containing the edges

$$\left\{e^1_1, \ldots, e^1_{k_1}, \ldots, e^n_1, \ldots, e^n_{k_n}, e_1, \ldots, e_n\right\}.$$

Therefore, $\{w \in \mathcal{W}_T : w > 0\} = \{d_i : (0, d_i) \in D_{<\infty}\}$. Since the multisets of weights of all minimum spanning trees of G coincide, the equality is satisfied for any minimum spanning tree T of G, as claimed. ■

Appendix B. Differentiability of functions on persistence diagrams

In this appendix, we prove Theorem 1 using methods and results from Leygonie et al. (2022). We consider finite ordered sets $X \subseteq \mathbb{R}^n$ with c elements, where $n \geq 2$ and $c \geq 2$. Each such set X corresponds to an element $(p_1, \ldots, p_c) \in \mathbb{R}^{cn}$, where $p_i \in \mathbb{R}^n$ denotes the ith point of X.

Given points $p_1, \ldots, p_c$ in $\mathbb{R}^n$, where $c \geq 2$, the *covariance* between p_i and p_j is

$$\mathrm{cov}(p_i, p_j) = \frac{1}{n} \sum_{k=1}^{n} (p_{i,k} - \bar{p}_i)(p_{j,k} - \bar{p}_j),$$

where $\bar{p}_i = \frac{1}{n} \sum_{k=1}^{n} p_{i,k}$. Since the covariance function is polynomial on the entries, the set

$$\mathcal{D}_{c,n} = \{(p_1, \ldots, p_c) \in \mathbb{R}^{cn} : \mathrm{cov}(p_i, p_j) \neq 0 \text{ for all } i, j \in \{1, \ldots, c\}\}$$

is open and dense in $\mathbb{R}^{cn}$.

Suppose given a dissimilarity $d \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$. A persistence diagram in homological dimension k can be viewed as a function $B_k = \mathrm{Dgm}_k \circ F$ defined on $\mathbb{R}^{cn}$ taking values in

the set **Diag** of families of points with multiplicities in the upper quadrant of $\mathbb{R}^2$ extended with points at infinity:

$$\mathbb{R}^{cn} \xrightarrow{F} \mathbb{R}^K \xrightarrow{\mathrm{Dgm}_k} \mathbf{Diag}. \tag{5}$$

Here we denote by $\mathbb{R}^K$ the set of all functions $f\colon K \to \mathbb{R}$, where K is the collection of nonempty subsets of $\{1,\dots,c\}$, which we view as faces of a $(c-1)$-dimensional simplex. The function F is defined as

$$F(X)(\sigma) = \max_{i,j\in\sigma} d(p_i, p_j)$$

for $\sigma \in K$ and $X = \{p_1,\dots,p_c\}$. The function Dgm_k assigns to each function $f\colon K \to \mathbb{R}$ the corresponding Vietoris–Rips persistence diagram in homological dimension k, where f is treated as a filtering function on the faces of a $(c-1)$-dimensional simplex.

Differentiability of functions valued in **Diag** is defined in Leygonie et al. (2022) as follows. For $m,\ell \in \mathbb{Z}_{\geq 0}$, consider the quotient map $Q_{m,\ell}\colon \mathbb{R}^{2m}\times\mathbb{R}^{\ell} \to \mathbf{Diag}$ sending each point

$$\tilde{D} = (x_1, y_1, \dots, x_m, y_m, z_1, \dots, z_\ell) \in \mathbb{R}^{2m}\times\mathbb{R}^{\ell}$$

to the diagram obtained by forgetting the order of the points:

$$Q_{m,p}(\tilde{D}) = \{(x_i,y_i)\}_{i=1}^{m} \cup \{(z_j,\infty)\}_{j=1}^{\ell} \cup \Delta^{\infty}.$$

Let $\mathcal{M}$ be a smooth manifold and let $B\colon \mathcal{M} \to \mathbf{Diag}$ be any map. For $x \in \mathcal{M}$ and $r \in \mathbb{Z}_{\geq 0}\cup\{\infty\}$, the map B is said to be *r-differentiable* at x if there exists an open neighborhood U of x and there exist integers $m,\ell \in \mathbb{Z}_{\geq 0}$ and a map $\tilde{B}\colon U \to \mathbb{R}^{2m}\times\mathbb{R}^{\ell}$ of class $\mathcal{C}^r$ such that $B = Q_{m,\ell}\circ\tilde{B}$ on U. Similarly, for a smooth manifold $\mathcal{N}$, a map $V\colon \mathbf{Diag} \to \mathcal{N}$ is said to be *r-differentiable* at a diagram D, where $r \in \mathbb{Z}_{\geq 0}\cup\{\infty\}$, if for all $m,\ell \in \mathbb{Z}_{\geq 0}$ and all $\tilde{D} \in \mathbb{R}^{2m}\times\mathbb{R}^{\ell}$ such that $Q_{m,\ell}(\tilde{D}) = D$ the map $V\circ Q_{m,\ell}\colon \mathbb{R}^{2m}\times\mathbb{R}^{\ell} \to \mathcal{N}$ is $\mathcal{C}^r$ on an open neighborhood of $\tilde{D}$.

As proved in Leygonie et al. (2022, Proposition 3.14), if a function $B\colon \mathcal{M} \to \mathbf{Diag}$ is r-differentiable at $x \in \mathcal{M}$ and another function $V\colon \mathbf{Diag} \to \mathcal{N}$ is r-differentiable at $B(x)$, then $V\circ B\colon \mathcal{M} \to \mathcal{N}$ is $\mathcal{C}^r$ at x as a map between smooth manifolds.

In what follows, we consider the projections for $i,j \in \{1,\dots,c\}$,

$$\pi_{i,j}\colon \mathbb{R}^{cn} \to \mathbb{R}^n\times\mathbb{R}^n, \qquad \pi_{i,j}(p_1,\dots,p_c) = (p_i,p_j).$$

Proposition 3 *Let $d\colon \mathbb{R}^n\times\mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a dissimilarity which is $\mathcal{C}^r$ on an open set $U \subseteq \mathbb{R}^n\times\mathbb{R}^n$, where $r \geq 0$. Let $p = (p_1,\dots,p_c) \in \mathbb{R}^{cn}$ such that $p \in \pi_{i,j}^{-1}(U)$ for all $i,j \in \{1,\dots,c\}$. Suppose that $d(p_i,p_j) \neq d(p_k,p_l)$ when $\{i,j\} \neq \{k,l\}$, where $i,j,k,l \in \{1,\dots,c\}$. Then the function $B_k = \mathrm{Dgm}_k \circ F$ defined in (5) is r-differentiable at p.*

Proof Since $d(p_i,p_j) \neq d(p_k,p_l)$ for $\{i,j\} \neq \{k,l\}$, the values $d(p_i,p_j)$ for $i \neq j$ are strictly ordered. As the projections $\pi_{i,j}$ are $\mathcal{C}^\infty$ and d is $\mathcal{C}^r$ on U, and $\pi_{i,j}(p) \in U$, we infer that $d\circ\pi_{i,j}$ is $\mathcal{C}^r$ in p. Since, in particular, $d\circ\pi_{i,j}$ is continuous, there is a neighbourhood U' of p where the order of the values $d(p'_i,p'_j)$ remains the same. Then $F(p)$ and $F(p')$ induce the same preorder on the set of simplices of K for every $p' \in U \cap \bigcap_{i,j}\pi_{i,j}^{-1}(U)$. Hence, the hypotheses of Leygonie et al. (2022, Theorem 4.7) hold and B_k is r-differentiable at p. ■

In what follows, we denote, for a given dissimilarity d,

$$\mathfrak{D}_{c,n} = \{(p_1, \ldots, p_c) \in \mathcal{D}_{c,n} : d(p_i, p_j) \neq d(p_k, p_l) \text{ if } \{i, j\} \neq \{k, l\}\}.$$

We note that, if the dissimilarity $d(x, y) = 1 - |\mathrm{corr}(x, y)|$ is chosen, then $\mathfrak{D}_{c,n}$ is an open dense subset of $\mathbb{R}^{cn}$, since $d(p_i, p_j) = d(p_k, p_l)$ precisely when $|\mathrm{corr}(p_i, p_j)| = |\mathrm{corr}(p_k, p_l)|$, and the square of correlation is a rational function.

Proposition 4 *Let $c, n \geq 2$, $k \in \mathbb{Z}_{\geq 0}$, and $d\colon \mathbb{R}^n \times \mathbb{R}^n \to [0, 1]$ be the dissimilarity given by $d(x, y) = 1 - |\mathrm{corr}(x, y)|$. The function $B_k = \mathrm{Dgm}_k \circ F$ in (5) is ∞-differentiable on $\mathfrak{D}_{c,n}$.*

Proof Take $p = (p_1, \ldots, p_c) \in \mathfrak{D}_{c,n}$. By the definition of $\mathfrak{D}_{c,n}$, we have that $d(p_i, p_j) \neq d(p_k, p_l)$ for all $\{i, j\} \neq \{k, l\}$. Furthermore, the correlation is well-defined and $\mathcal{C}^\infty$ on $\pi_{i,j}(p) = (p_i, p_j)$ for all $i, j \in \{1, \ldots, c\}$, because $\mathrm{cov}(p_i, p_j) \neq 0$ for points in $\mathfrak{D}_{c,n}$. We also have that $|\mathrm{corr}(x, y)|$ is $\mathcal{C}^\infty$ on every (p_i, p_j), since the absolute value function is $\mathcal{C}^\infty$ on $\mathbb{R} \smallsetminus \{0\}$. Therefore, d is $\mathcal{C}^\infty$ on $\pi_{i,j}(p) = (p_i, p_j)$ for all $i, j \in \{1, \ldots, c\}$ and thus the assumptions of Proposition 3 hold, implying that B_k is ∞-differentiable on $\mathfrak{D}_{c,n}$. ∎

Proof of Theorem 1 By Proposition 4, the function B_0 is ∞-differentiable on $\mathfrak{D}_{c,n}$, where $B_0(X)$ is the persistence diagram of X in homological dimension zero. Therefore, we only need to display functions $\tilde{\mathcal{T}}_i\colon \mathbf{Diag} \to \mathbb{R}$ that are ∞-differentiable on $B_0(\mathfrak{D}_{c,n})$ such that $\mathcal{T}_i = \tilde{\mathcal{T}}_i \circ B_0$ for $i \in \{1, 2\}$.

Here we view zero-dimensional persistence diagrams as consisting of points $(0, y)$, although we keep denoting them in the general form (b, d). When computing the average persistence and standard deviation of persistence of the points in $B_0(X)$, the number of points in the diagram is assumed to be equal to the number of edges of a minimum spanning tree for (X, d), that is, $|X| - 1$. Therefore, the functions $\tilde{\mathcal{T}}_i$ can be defined as

$$\tilde{\mathcal{T}}_1(D) \triangleq \frac{1}{c-1} \sum_{(b,d) \in D^*} (d - b), \qquad \tilde{\mathcal{T}}_2(D) \triangleq -\alpha \tilde{\mathcal{T}}_1(D) + \beta \sigma(D),$$

where

$$\sigma^2(D) \triangleq \frac{1}{c-1} \sum_{(b,d) \in D^*} ((d - b) - \tilde{\mathcal{T}}_1(D))^2$$

with $D^* = \{(b, d) \in D : d < \infty\}$. Points in the diagonal Δ^∞ are sent to zero by $d - b$ and are not taken into consideration in the sums, and neither are points at infinity.

In order to prove that the functions $\tilde{\mathcal{T}}_i$ are ∞-differentiable, take any $m, \ell \in \mathbb{Z}_{\geq 0}$ and $\tilde{D} \in \mathbb{R}^{2m} \times \mathbb{R}^\ell$ such that $Q_{m,\ell}(\tilde{D}) = D$. If we write $\tilde{D} = (x_1, y_1, \ldots, x_m, y_m, z_1, \ldots, z_\ell)$, then the functions $\tilde{\mathcal{T}}_i \circ Q_{m,\ell}$ are given by

$$\tilde{\mathcal{T}}_1(Q_{m,\ell}(\tilde{D})) = \frac{1}{c-1} \sum_{i=1}^{m} (y_i - x_i), \qquad \tilde{\mathcal{T}}_2(Q_{m,\ell}(\tilde{D})) = -\alpha \tilde{\mathcal{T}}_1(Q_{m,\ell}(\tilde{D})) + \beta \sigma(\tilde{D}),$$

where

$$\sigma^2(\tilde{D}) \triangleq \frac{1}{c-1} \sum_{i=1}^{m} [(y_i - x_i) - \tilde{\mathcal{T}}_1(Q_{m,\ell}(\tilde{D}))]^2.$$

The functions $\tilde{\mathcal{T}}_i \circ Q_{m,\ell}$ are $\mathcal{C}^\infty$ on all their domain because they are compositions of $\mathcal{C}^\infty$ functions on a neighborhood of $\tilde{D}$. The only function that is not $\mathcal{C}^\infty$ in all its domain is the square root function, which is not differentiable at zero. However, for points $p \in \mathfrak{D}_{c,n}$ we have pairwise different distances, and consequently the persistence diagram $B_0(X)$ contains at least two different points, making $\sigma(\tilde{D}) \neq 0$ for a neighbourhood U of $\tilde{D}$ and thus making $\sigma(\tilde{D})$ a $\mathcal{C}^\infty$ function on U. Hence, $\tilde{\mathcal{T}}_i$ is ∞-differentiable. Therefore, as B_0 and $\tilde{\mathcal{T}}_i$ are ∞-differentiable in $\mathfrak{D}$ and $B_0(\mathfrak{D}_{c,n})$ respectively, the functions $\mathcal{T}_i$ are $\mathcal{C}^\infty$ on $\mathfrak{D}_{c,n}$. ■

Appendix C. Resources used and computation

The experiments were computed in a server with 503 GB of RAM, a CPU AMD EPYC 7452 32-Core Processor with a frequency up to 3.35 GHz, and seven GPUs NVIDIA GeForce RTX 3090 with 24 GiB of memory. The storage consisted of 3 Samsung SSDs, two of them with 3840 GB of memory and the other one with 960 GB. All the experiments were executed in parallel using one of the GPUs per experiment. The computational bottlenecks were related to the computation of correlation matrices of neurons and persistence diagrams. The first process was done using TensorFlow (in GPU mode) and the second one was performed using the library `giotto-ph` (Pérez et al., 2021).

Appendix D. Figures

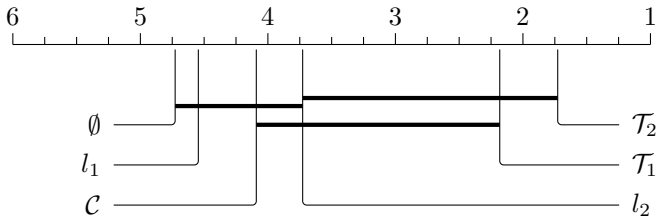

Figure 1: Critical difference diagrams (Demšar, 2006) for the Friedman and Nemenyi post-hoc statistical tests conducted in both blocks of experiments. The position of each training approach on the diagram corresponds to its average rank based on the test accuracies of the trained network —see Table 1. Lower ranks indicate that a regularisation term, or the training without regulariser, outperforms competitors with higher ranks. Regularisation terms are connected if the p-value obtained from the Nemenyi post-hoc test is greater than 0.05. See Table 2 for a description of the training approaches and the p-values.