node is augmented with some feature visualization techniques or statistical data to obtain more insight into the activations that comprise it.

One of the visualization techniques proposed by Rathore et al. was the computation of *activation images* for the activations collected in each node and for its average. Activation images, given an activation vector or an average of them, are *idealized* images that would have produced the activations via an iterative optimization process proposed by Olah et al. (2017, 2018). Other statistical data proposed by Purvine et al. (2023) to describe Mapper nodes were: 1. Pie charts showing the composition of class labels in that node, associating to each activation vector the label of the image that generated the activation vector; 2. Node–wise purity, defined as $\alpha_i = c_i^{-1}$ where c_i is the number of class labels in the Mapper node i; 3. Class-wise purity, defined as $\gamma_y = |J_y|^{-1} \sum_{x \in J_y} \beta_x$, where J_y is the set of activations in the Mapper graph associated to the class y and $\beta_x = |I_x|^{-1} \sum_{i \in I_x} \alpha_i$, with I_x being the set of nodes in the Mapper graph containing the activation x.

The construction method for activation vectors varies depending on the type of layer under consideration. In particular, the convolutional layer emerged as the most extensively studied using TopoAct among the articles analyzed in this survey. For convolutional layers with output values of dimension $h \times w \times c$, Rathore et al. (2021) proposed to compute activations for each input value by randomly selecting two indices $i, j \in [h] \times [w]$ and taking the c dimensional vector obtained by fixing the dimensions i, j of the output. Later, two other sampling methods for activation vectors were proposed by Purvine et al. (2023): 1. Sampling all the possible $h \cdot w$ activations vectors for all possible pairs of indices i, j; 2. Sampling activations such that the indices i, j whose receptive fields are associated with the most quantity of background or foreground pixels in the inputs are the ones selected.

For pretrained versions of InceptionV1 (Szegedy et al., 2015), BERT (Devlin et al., 2019), and ResNet-18 (He et al., 2016) the Mapper graphs generated by TopoAct carried meaningful information. For InceptionV1, Rathore et al. observed that branches in the Mapper graphs corresponded to activations containing different features of the inputs. For example, the activation images for one bifurcation showed that the nodes of one branch were associated with animal legs, while the nodes of the other were associated with distorted faces. Similarly, loops of nodes were associated with different aspects and features of the same underlying objects. An example of this was found in a loop that contained six nodes, each representing different features (body parts) of some set of animals, including dogs and foxes. For BERT, a language representation model, the Mapper graphs showed a similar behavior separating word representations, where branches seemed to separate contextual meaning of similar words.

For ResNet-18, Rathore et al. observed branching patterns similar to those seen in the InceptionV1 model, suggesting that TopoAct results are not specific to a particular dataset or architecture. Furthermore, Purvine et al. observed that the complete and random activation sampling methods yielded similar bifurcation patterns. For the sampling method associated with the most background or foreground pixels in the input images, notable class bifurcations, that is, branches in the graph separating activations associated with different labels, seemed to appear at earlier layers, although this is not evident from the experiments. In general, node-wise and class-wise purity were found to be higher in deeper layers for all the sampling methods, confirming the idea that models get better at separating the classes the deeper they go.

In a more quantitative way, Purvine et al. also proposed a dissimilarity measure to quantitatively compare two layers. This dissimilarity is given by the sliced Wasserstein distances of the persistence diagrams induced by samples of the activations of both layers. Specifically, the activations with the highest l^2 norms are taken. However, although the measure was robust to different weight initializations, the dissimilarity did not pass some *sensitivity tests* proposed by (Ding et al., 2021) that reasonable measures between layers should pass.

TopoAct has also been used by Zhou et al. (2023) to analyze the effect of adversarial examples in neural networks using the Mapper graphs of one of their deepest layers computed with the training dataset. Specifically, Mapper graphs are studied using a variant of the node-wise purity introduced in Rathore et al. (2023). For a node i, the purity of i is $1 - H(D_i)/H(D)$ where H denotes the Shannon entropy of a distribution, D_i denotes the observed distribution of labels for the points in the node i, and D denotes a uniform distribution of all labels. This purity reaches one whenever all points in X are of the same class and zero when points are distributed uniformly over all classes. Zhou et al. studied how performing adversarial trainings, i.e., training neural networks with adversarial examples, affects the configuration of the Mapper graph. The experiments were performed in two scenarios: 1. Training a simple FCFNN model with MNIST; 2. Training a ResNet-18 with CIFAR-10.

In the first case, where no overfitting was observed during training, impure nodes, that is, nodes that do not contain a dominant label in them, of the neural network trained without adversarial examples captured decision boundaries. Also, the higher the attack, that is, the bigger the perturbations made to the original examples, the higher the number of nodes with low accuracy in the Mapper graphs, i.e., the higher the number of nodes containing activations coming from inputs that were misclassified.

In the second case, where overfitting was observed, the higher the attack, the lower the weighted average purity, and the lower the test accuracy, where the purity is weighted by the number of activations in each node divided by the total number of activations in the Mapper graph. The observation that the model is overfitting and that the purity decreases with the attack suggests that, in this case, the impure nodes were also capturing decision boundaries.

Based on such observations, Zhou et al. propose to improve the robustness of adversarially trained neural networks by selecting misclassified activations from low-accuracy nodes on the mapper graph and using them to refine the model. For the FCFNN network in MNIST, this procedure marginally improved the accuracy of the model. However, for the ResNet-18 in CIFAR-10, this procedure had no effect.

3.3.2 Homology and persistent homology

The literature has a wealth of methods to leverage the information provided by (persistent) homology features induced by activations and weights of neural networks. We divide this section into six blocks, each representing a space of internal representations in which TDA is applied, as follows: 1. Activations in the complete neural network graph; 2. Activations for each layer; 3. Weights in the complete neural network graph; 4. Weights layer by layer; 5. Activations whose dissimilarities depend on the weights; and 6. Generic spaces.

Activations in the complete neural network graph

The (persistent) homology of activation vectors has been successfully used to analyze many aspects of deep neural networks, such as their generalization or interpretability. In an early study within this section, Corneanu et al. (2019) analyzed how diverse topological information extracted from the set of neuron activations of neural networks was correlated with their generalization capabilities. The neuron activations were studied for the complete graph at the same time, and the activation vector $a_{v_i^l}$ for each neuron v_i^l was taken as

$$a_{v_i^l} = \left(\phi_{\mathcal{N}}^{(l)}(x_1)_i, \ldots, \phi_{\mathcal{N}}^{(l)}(x_n)_i\right),$$

for a fixed set of inputs $\mathcal{D} = \{x_i\}_{i=1}^n$ to the neural network $\mathcal{N}$, in this article a (sub)set of the training dataset.

In their first experiments, Corneanu et al. studied how the number of simplices of Vietoris–Rips simplicial complexes, given by the formula

$$S(n) = |\sigma \in \mathrm{VR}_t(P, d_\downarrow) : \dim(\sigma) = n|,$$

varied during several training processes of a LeNet neural network architecture (Lecun et al., 1998). Given a neural network $\mathcal{N}$ and a fixed value T, Vietoris–Rips simplicial complexes were computed from point clouds $(P, d_\downarrow)$ induced from a weighted connected graph $\mathcal{F}_\mathcal{N}$ such that: 1. The vertices are a subset of neurons of $\mathcal{N}$; 2. Edges are weighted by $w_E(\{v, w\}) = |\mathrm{corr}(a_v, a_w)|$, where corr is the sample Pearson correlation and a_v and a_w are the activation vectors of the neuron v and w, respectively; 3. An edge $\{v, w\}$ is in $E(\mathcal{F}_\mathcal{N})$ if $w_E(\{v, w\}) > T$; 4. Only edge endpoints are added to the vertex set. The parameter t for the Vietoris–Rips simplicial complexes was chosen such that the density ρ_t of the edges with non-zero correlation for the parameter t, given by

$$\rho_t = \frac{|\{\sigma = \{v, w\} \in \mathrm{VR}_t(P, d_\downarrow) : \dim(\sigma) = 1 \text{ and } |\mathrm{corr}(a_v, a_w)| > 0\}|}{\{\sigma = \{v, w\} \in |E(\mathcal{F}_\mathcal{N}) : |\mathrm{corr}(a_v, a_w)| > 0|\}},$$

was near to 0.25. In the experiments, it was found that, the higher the area under the curve of $S(n)$, the better the generalization capabilities of the neural network. In particular, it was shown that, for such training procedures, the function $S(n)$ is capable of distinguishing between the three main regimes given during training: underfitting, generalization, and overfitting. That is, the training process starts with a small area under $S(n)$ (underfitting regime), then $S(n)$ grows and achieves its maximum value (generalization regime), and then decreases again (overfitting regime).

The previous results motivated more involved experiments to see the relationship between the topological properties of the activations of neural networks and their generalization capacity. In particular, a new functional graph $\mathfrak{F}_\mathcal{N}$ is built to induce Vietoris–Rips filtrations to extract sharper topological information about the network. In this case, the functional graph is complete, its set of vertices is a fixed (sub)set of neurons of $\mathcal{N}$, and its weights are given by a correlation dissimilarity $w_E(\{v, w\}) = 1 - |\mathrm{corr}(a_v, a_w)|$. Here $\mathfrak{F}_\mathcal{N}$ induces point clouds (P, d) of neurons where $P = V(\mathfrak{F}_\mathcal{N})$ and $d = w_E$. From this point cloud (P, d), generated again during several training procedures of a LeNet neural network, Corneanu et al. computed persistence diagrams $D(\mathbb{V}_k(\mathrm{VR}(P, d)))$ for $k \in \{1, 2, 3\}$ and a