models in an ensemble method used in an appropriate way. Finally, they found that the representation topology divergence comparing the trained models correlated well with the disagreement of the predictions, although more experiments are needed to understand this relationship.

3.4 Training dynamics and loss functions

In this section, we review only three articles that focus on studying the properties of the training process. The first one deals with the loss function used to train the neural network. The other two are focused on the evolution of the weights during the training process and how the *fractal dimension* of weight trajectories are related to the generalization capacity of neural networks.

Loss functions

One of the fundamental problems in deep learning is, given a learning problem, an architecture a, a loss function $\mathcal{L}$, and a training algorithm $\mathcal{A}$, to determine if the training algorithm $\mathcal{A}$ is capable of finding a neural network $\mathcal{N}$ with architecture a that minimizes the empirical risk $\widehat{\mathcal{R}}_{\mathcal{D}_{\text{train}}}$ for the learning problem. Usual training algorithms perform a gradient descent, following a path in the space of parameters of the neural network. This path is then heavily affected by the connectivity of the loss graph and by the presence of *local valleys*, in which the gradient descent algorithm can get stuck.

Nguyen (2019) studied a generalization of this problem for general FCFNNs depending on their activation functions, their graph structure (depth and width of layers), and the *shape* of the training dataset. Specifically, he studied the number of connected components and the existence of local valleys on the graph of an optimization target $\mathfrak{L}(\theta) = f(\phi_{\mathcal{N}_\theta}(x_1), \dots, \phi_{\mathcal{N}_\theta}(x_m))$, where f is a convex function, and $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^m$. These different optimization targets allow training algorithms to minimize any convex function with respect to the parameters of neural networks that could be useful to obtain better parameters for the network, not restricting the minimization to empirical risks. Although this is a more complex scenario than the one presented in Section 2.2, empirical risk functions for many loss functions, such as categorical cross entropy, are examples of these types of optimization targets. For the discussion of this article, we assume that the activation functions for all layers except for the last one are the same, and that the last-layer activation function is simply the identity.

The first insight on the optimization target graph is that, for strictly monotonic activation functions φ with $\text{Im}(\varphi) = \mathbb{R}$, FCFNNs with widths strictly decreasing layer by layer, that is, $N_i > N_{i+1}$ for $i \in [L-1]$ and with at least two non-input layers, i.e., $L \geq 2$, and a linearly independent training dataset $\mathcal{D}_{\text{train}}$, every sublevel set of $\mathfrak{L}$, that is, the sets $\mathfrak{L}^{-1}(-\infty, \alpha]$ for $\alpha \in \mathbb{R}$, is connected and also every non-empty connected component of every level set $\mathfrak{L}^{-1}(\alpha)$ is unbounded. Connectedness of sublevel sets implies a well-behaved optimization target function, and unboundedness of level sets implies that there are no local valleys in the optimization target graph, understanding by a local valley a non-empty connected component of some strict sublevel set $\mathfrak{L}^{-1}(-\infty, \alpha)$.

A bad local valley is a local valley in which the target $\mathfrak{L}$ cannot be arbitrarily close to the lower bound of the convex function inducing the target function, which is also a lower bound of $\mathfrak{L}$ but does not depend on any concrete neural network. Bad local valleys

are harmful to the optimization problem because if the training process enters them when optimizing the target function, the parameters obtained at the end of the learning process are provably not optimal. Nguyen proved that, for activation functions as before that also satisfy that there are no nonzero coefficients $(\lambda_i, a_i)_{i=1}^p$ with $a_i \neq a_j$ for all $i \neq j$ such that $\varphi(x) = \sum_{i=1}^p \lambda_i \varphi(x - a_i)$ for all $x \in \mathbb{R}$, FCFNNs with a layer $l \in [L-1]$ satisfying $N_l > |\mathcal{D}_{\text{train}}|$ and $N_i > N_{i+1}$ for $i \in \{l+1, \ldots, L\}$ induce target functions $\mathfrak{L}$ with no bad local valleys and, if $l \leq L-2$, then every local valley of $\mathfrak{L}$ is unbounded. This implies that from any initial parameter, there is a continuous path on which the loss function is nonincreasing from it to a point that is arbitrarily close to the infimum of the loss.

As we have seen so far, widths of neural network layers play a significant role in the configuration of the optimization target graph. The first hidden layer always plays an important role in this configuration, since it determines the first transformation of the data into some space from which features from the inputs are extracted. Assuming that the activation functions of the network satisfy the two previous assumptions, and assuming that $N_1 > 2\,|\mathcal{D}_{\text{train}}|$ and that $N_i > N_{i+1}$ for $i \in \{2, \ldots, L-1\}$, Nguyen proved that every sublevel set of $\mathfrak{L}$ is connected and also that every connected component of every level set of $\mathfrak{L}$ is unbounded. This is a stronger result than the previous one, as it implies that not only there are no bad local valleys but also there is a unique global valley.

Many current activation functions, such as the leaky ReLU, satisfy the previous assumptions. However, the usual ReLU does not and further assumptions are needed to have a good behavior of the target graph. Letting $N_{\min} = \min_{i \in [L-1]} N_i$, for FCFNNs with activation functions φ only satisfying the last assumption implying coefficients $(\lambda_i, a_i)_{i=1}^p$, if $N_{\min} > |\mathcal{D}_{\text{train}}|$, then $\mathfrak{L}$ has no bad local valleys and, if $N_{\min} > 2\,|\mathcal{D}_{\text{train}}|$, then every sublevel set of $\mathfrak{L}$ is connected.

Some of the previous assumptions are too strong to be satisfied in many practical scenarios. However, these results serve as a basis for a better understanding of the shape of the target function. A similar approach using topological data analysis to capture the connectivity and shape of optimization target graphs for modern neural networks could be a promising research line. This could help verify if the claims given in these cases can be extrapolated to more general scenarios.

Fractal dimension of weight trajectories

The last approach that we discuss in this survey is related to the evolution of parameters during the training process. In this case, training algorithms $\mathcal{A}$ are assumed to be continuous, which means that the parameters evolve over a time period $[0, T] \subseteq \mathbb{R}$ for which each instant of time$t \in [0, T]$ has an associated weight value θ_t. Although this is not the case for real scenarios, since computers work in the discrete domain, there are good continuous approximations of classical *discrete* optimization algorithms such as gradient descent that allow us to study these discrete processes with properties of the continuous approximations.

Simsekli et al. (2020) discovered that the generalization capacity of neural networks was connected with the heavy-tailed behavior of weight trajectories $\Theta_{\mathcal{A}} = \{\theta_t : t \in [0, T]\}$ generated during training, in particular, with its upper box dimension, a dimensionality measure for fractals. A difficulty with this link is that many strong assumptions about the training algorithm and the space of weights were needed to formally prove the connection.

Birdal et al. (2021) relaxed these assumptions by cleverly using a previous result that connected this fractal dimension with persistent homology (Kozma et al., 2006; Schweinhart, 2021). This result says that, for $\Theta \subseteq R^d$ a bounded set, we have

$$\dim_{\mathrm{PH}}\Theta = \dim^0_{\mathrm{PH}}\Theta = \dim_{\mathrm{Box}}\Theta,$$

where $\dim_{\mathrm{Box}}$ is the upper-box fractal dimension, and $\dim^k_{\mathrm{PH}}$ is the persistent homology dimension given by

$$\dim^k_{\mathrm{PH}}\Theta = \inf\left\{\alpha : E^k_\alpha(\Theta_{<\infty}) < C : \exists C > 0 \text{ for all finite } \Theta_{<\infty} \subseteq \Theta\right\},$$
$$E^k_\alpha(P) = \sum_{(b,d)\in D} |d-b|^\alpha, \qquad D = D(\mathbb{V}_k(\mathrm{VR}(P, \|\cdot\|_2))),$$

that is, the infimum of all the exponents for which E^k_α is uniformly bounded for all finite subsets Θ_{inf}.

In particular, Birdal et al. proved that, given any compact set of (random) possible weights Θ, for example, the weight training trajectories $\Theta_{\mathcal{A}}$, $\mathcal{L}(f, x_i, y_i)$, and a loss function $\mathcal{L}$ bounded by B and K-Lipschitz continuous on the set of possible parameters θ of a fixed neural network architecture, then, for a sufficiently large size of the training dataset m, the following holds:

$$\sup_{\theta\in\Theta}\left|\widehat{\mathcal{R}}_{\mathcal{D}_{\mathrm{train}}}(\phi_{\mathcal{N}_\theta}) - \mathcal{R}(\phi_{\mathcal{N}_\theta})\right| \leq 2B\sqrt{\frac{(\dim_{\mathrm{PH}}\Theta + 1)\log^2(mL^2)}{m} + \frac{\log(7M/\gamma)}{m}}$$

with probability at least $1-\gamma$ over a training dataset $\mathcal{D}_{\mathrm{train}}$ with m elements sampled i.i.d. from the data distribution, where $\mathcal{N}_\theta$ denotes the neural network $\mathcal{N}$ defined with the fixed architecture and parameters θ and M denotes a constant depending on some technical assumptions about the objects involved in the bound.

The previous bound uses persistent homology dimension, which cannot be computed exactly by a computer program. Birdal et al. propose an algorithm to estimate this quantity for finite sets of weights. Estimations of persistent homology dimension were found to be significantly correlated with the generalization gap calculated as the difference between the accuracy in train and the accuracy in test, indicating that lower persistent homology dimensions were associated with better generalization capacities of networks for a wide variety of networks, including simple FCFNN models and AlexNet, trained on MNIST, CIFAR-10, and CIFAR-100 datasets.

Finally, the differentiability theory for persistence diagrams allows one to minimize these estimations as a regularization method, expecting to obtain better generalizations for regularized models. This is exactly what happened for a LeNet-5 (Lecun et al., 1998) architecture trained on CIFAR-10 in the experiments performed by Birdal et al., especially for training procedures that failed to converge to a good set of parameters only by themselves.

The previous bounds were improved in a follow-up paper by Dupuis et al. (2023) in which the persistent homology dimension was used again to compute fractal dimensions. As the methods employed to improve the bounds are not related with topological data analysis, we do not analyze it in this survey, although the results are very insightful and interesting in their own right.