Journal of Machine Learning Research 26 (2025) 1-53 Submitted 12/20; Revised 11/25; Published 11/25

Almost Sure Convergence of Dropout Algorithms for Neural Networks

Albert Senen–Cerda ALBERT.SENEN-CERDA@POSTEO.NET
Jaron Sanders JARON.SANDERS@TUE.NL
Department of Mathematics & Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands

Editor: Pradeep Ravikumar

Abstract

We investigate the convergence and convergence rate of stochastic training algorithms for Neural Networks (NNs) that have been inspired by *Dropout* (Hinton et al., 2012). With the goal of avoiding overfitting during training in NNs, dropout algorithms consist in practice of multiplying the weight matrices of a NN componentwise by independently drawn random matrices with $\{0, 1\}$-valued entries during each iteration of Stochastic Gradient Descent (SGD). This paper presents a probability theoretical proof that for fully-connected NNs with differentiable, polynomially bounded activation functions, if we project the weights onto a compact set when using a dropout algorithm, then the weights of the NN converge to a unique stationary point of a projected system of Ordinary Differential Equations (ODEs).

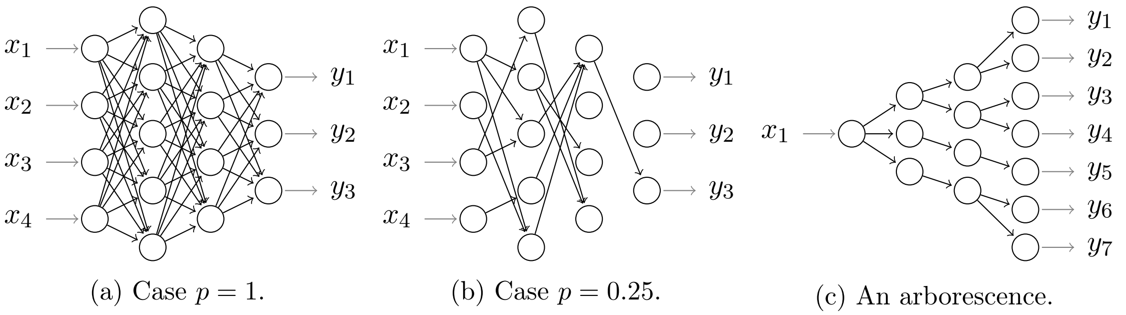
After this general convergence guarantee, we go on to investigate the convergence rate of dropout. Firstly, we obtain generic sample complexity bounds for finding ϵ-stationary points of smooth nonconvex functions using SGD with dropout that explicitly depend on the dropout probability. Secondly, we obtain an upper bound on the rate of convergence of Gradient Descent (GD) on the limiting ODEs of dropout algorithms for NNs with the shape of an arborescence of arbitrary depth and with linear activation functions. The latter bound shows that for an algorithm such as *Dropout* or *Dropconnect*(Wan et al., 2013), the convergence rate can be impaired exponentially by the depth of the arborescence.

In contrast, we experimentally observe no such dependence for wide NNs with just a few dropout layers. We also provide a heuristic argument for this observation. Our results suggest that there is a change of scale of the effect of the dropout probability in the convergence rate that depends on the relative size of the width of the NN compared to its depth.

Keywords: dropout, convergence, neural networks, stochastic approximation, ODE method

1. Introduction

Dropout (Hinton et al., 2012) is a technique to avoid overfitting during training of NNs that consists of temporarily ‘dropping’ nodes of the network independently at each step of SGD. While in the original *Dropout* algorithm in Hinton et al. (2012) only nodes from the network were dropped, several stochastic training algorithms that avoid overfitting in NNs have appeared since then; for example, *Dropconnect* (Wan et al., 2013), *Cutout* (DeVries and Taylor, 2017). Figure 1 depicts a NN where we use *Dropconnect* and drop individual edges instead of nodes. In practice, such dropout algorithms consist of multiplying componentwise weight matrices of the NN in each iteration by independently drawn random matrices with $\{0, 1\}$-valued entries. The elements of these random matrices indicate whether each individual edge or node is filtered (0) or is not filtered (1) during a training step. The resulting weight matrices are then used in the backpropagation algorithm for computing the gradient of a NN. Mathematically, dropout turns the backpropagation algorithm into a step of a SGD in which the primary source of randomness is the NN's configuration. Under mild independence assumptions, the loss function of dropout is a risk function averaged over all possible NNs configurations (Baldi and Sadowski, 2013).



(a) Case $p = 1$.

(b) Case $p = 0.25$.

(c) An arborescence.

Figure 1: *(a,b) Dropconnect*'s training step (Wan et al., 2013) in a NN with $L = 3$ layers. In this algorithm, on every iteration, a random NN is first generated by removing each edge with probability $p \in (0, 1]$ independently of all other edges. The output of this random NN is then used to update all weights using the backpropagation algorithm. *(c)* An example arborescence of depth $L = 3$.

An interesting aspect of dropout algorithms is that they lie at the intersection of stochastic optimization and *percolation theory*, which investigates properties related to connectedness of random graphs and deterministic (possibly infinite) graphs in which vertices and edges are deleted at random. In the case of dropout, the output of the filtered NN with temporarily deleted edges is used to update the weights. If dropout filters too many weights, then little information about the input can pass through the network, which will consequently also yield a gradient update for that step that contains little relevant information.

As an example, we may consider again the networks in Figures 1 (a)–(b) when we use *Dropconnect*, that is, we filter each edge with probability $1 - p$ independently of all other edges. We can observe that the number of paths χ in Figure 1 (b) that fully transverse the network ($\chi = 5$) is much smaller compared to those of Figure 1 (a) ($\chi = 240$). In a NN with no biases and L dropout layers, a path from the input layer to the output goes through L weights that have filters. Then, the probability that a path from input to output stays unfiltered and contributes to a weight update is p^L. If we now fix one edge in the path, then the probability of updating its corresponding weight through that path in particular is also p^L. There are, however, many other paths in a NN passing through a single edge. The probability that one of those paths is not filtered will be large and may compensate the exponential factor p^L. Considering the connection to bond percolation, one may therefore suspect that dropout algorithms may perform worse than a routine implementation of the backpropagation algorithm. However, dropout algorithms usually perform well since they avoid overfitting in NNs (Hinton et al., 2012; Srivastava et al., 2014). From the point of view of bond percolation however, this should still come at the cost of slower convergence of dropout algorithms, and conceivably by as much as a factor p^L, where L is the number of dropout layers.

Most theoretical focus has been on the generalization properties of NNs trained with dropout algorithms. We can mention Hinton et al. (2012); Baldi and Sadowski (2013); Wager et al. (2013); Srivastava et al. (2014); Baldi and Sadowski (2014); Cavazza et al. (2018); Mianjy et al. (2018); Mianjy and Arora (2019); Pal et al. (2020); Wei et al. (2020), which we briefly review in Section 1.3. In this paper, however, we investigate dropout from the stochastic optimization perspective. That is, we aim to answer if dropout algorithms converge and study the rate at which they converge, which is expected to depend on the dropout probability. Compared to the study of the generalization properties of dropout, this aim has received less attention in the literature. In particular, we can only mention Mianjy and Arora (2020) and Senen-Cerda and Sanders (2022). In Mianjy and Arora (2020), a convergence rate for the test error in a classification setting is obtained when training shallow NNs with dropout. This rate, is, however, independent of the dropout probability. In Senen-Cerda and Sanders (2022), a convergence rate for the empirical risk associated with training shallow linear NNs with dropout is obtained that depends on the dropout probability. Both results refer to shallow NNs where the width of the NN plays a role in the convergence rate. We refer to Section 1.3 below for further details.

From the previous discussion, however, we suspect that there is an effect of dropout in the convergence rate in *deep* NNs with several layers of dropout. In this paper, we investigate this problem. In particular, we provide convergence guarantees for training NNs that have several layers of dropout and analyze simplified models for deep NNs, for which it is possible to obtain an explicit convergence rate that depends on the dropout probability and depth. We also consider the effect on the sample complexity of using dropout SGD and complement the previous results with simulations on realistic NNs to examine the convergence rate of dropout empirically.

Before introducing the results of the paper we briefly define the fundamental concepts related to training of NNs with dropout that we will use throughout this paper.

1.1 Dropout and SGD

A NN $\Psi_W : \mathcal{X} \to \mathcal{Y}$ with weights W is typically used to predict output $Y \in \mathcal{Y}$ given input $X \in \mathcal{X}$ both of which are sampled from some joint distribution. For a given loss $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the risk function of Ψ_W is usually defined as

$$\mathcal{U}(W) \triangleq \int l(\Psi_W(x), y)\, \mathrm{d}\mathbb{P}[(X,Y) = (x,y)], \tag{1}$$

where the distribution is usually given by the empirical distribution of a finite number of samples $\{(x_i, y_i)\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$. In this case, the risk is an *empirical risk*.

Ideally, the NN is operated using weights in the set $\arg\min_W \mathcal{U}(W)$. However, the weights are found in practice by using gradient descent or its stochastic variant SGD, which aims to minimize the risk in (1) by updating the weights in the local direction that minimizes the function. At time t, the weights $W^{[t]}$ of the NN are namely updated by setting

$$W^{[t+1]} = W^{[t]} - \alpha^{\{t+1\}} \tilde{\Delta}^{[t+1]}. \tag{2}$$

Here, $\tilde{\Delta}^{[t+1]}$ is a stochastic estimate of the gradient of (1) and $\alpha^{\{t+1\}}$ is a step size which we will specify later. Let $B_W(X, Y)$ be the gradient at W of (1). If the input and output samples $X^{[t+1]}, Y^{[t+1]}$ are provided at time t, then the update of SGD is given by

$$\tilde{\Delta}^{[t+1]} = B_{W^{[t]}}(X^{[t+1]}, Y^{[t+1]}). \tag{3}$$

As we have mentioned, dropout filters are applied to some of the weights W during training by using matrices of random variables F with $\{0,1\}$-valued entries. Denote by $F^{[t+1]}$, $X^{[t+1]}, Y^{[t+1]}$ the dropout filters and the samples provided to the SGD algorithm at time t, respectively. Compared to (3), a dropout algorithm defines the estimate of the gradient update as

$$\Delta^{[t+1]} \triangleq F^{[t+1]} \odot \mathrm{B}_{F^{[t+1]} \odot W^{[t]}}(X^{[t+1]}, Y^{[t+1]}), \tag{4}$$

where $\odot$ denotes the componentwise product.

Note that in (4) the filters appear twice. Firstly, they filter the weights $W^{[t]}$ when the gradient is computed depending only on the subnetwork provided by dropping some edges or nodes. Secondly, they filter the updates in $\Delta^{[t+1]}$ since only the remaining weights will be updated. We remark that in this general formulation, other distributions for the filters than those for dropout and dropconnect are allowed. For specific examples of distribution of the filter matrices we refer to Section 2.3.

We next present the results of this paper.

1.2 Summary of Results

Our first result is a formal probability theoretical proof that for any (fully connected) NN topology and with differentiable polynomially bounded activation functions (see Theorem 5), the iterates of projected SGD with dropout-like filters converge. In particular, a step of projected SGD with dropout is given by

$$W^{[t+1]} = \mathrm{P}_{\mathcal{H}}(W^{[t]} - \alpha^{\{t+1\}} \Delta^{[t+1]}) \quad \text{for} \quad t \in \mathbb{N}_0, \tag{5}$$

where $\Delta^{[t+1]}$ is the estimate of the gradient with dropout in (4) and $\mathrm{P}_{\mathcal{H}}$ is an operator that projects the iterates onto a compact convex set $\mathcal{H}$ (Oymak, 2018). In order to state our first result, we define a *dropout algorithm's risk function* as

$$\mathcal{D}(W) \triangleq \int l(\Psi_{f \odot W}(x), y) \, \mathrm{d}\mathbb{P}[(F, X, Y) = (f, x, y)], \tag{6}$$

and we will consider $l(a, b) = |a - b|^2$ to be the ℓ_2-loss. The result is stated informally in the next proposition.

Result 1 *(Informal statement of Proposition 6.) Under sufficient regularity of the activation functions, bounded moments and independence of random variables and some assumptions on the boundary $\mathcal{H}$, with update* (5)*, the weights $(W^{[t]})_t$ converge to a unique stationary set of a projected system of ODEs*

$$\frac{\mathrm{d}W}{\mathrm{d}t} = -\nabla_W \mathcal{D}|_{\mathcal{H}}(W) + \pi(W), \tag{7}$$

where $\pi(W)$ is a constraint term*, which describes the minimum vector required to keep the gradient flow of $\nabla \mathcal{D}$ in $\mathcal{H}$.*

This result provides a formal guarantee with the sufficient conditions for dropout algorithms to be well-behaved and at least asymptotically (meaning after sufficiently many iterations) to not suffer from problems that could have arisen from the relation to bond percolation. Moreover, for a wide range of NNs and activation functions the function $\mathcal{D}(W)$ is the expectation of the risk over the dropout's filters distribution, which in our result is not restricted to dropping nodes and can even be coupled to the data. This result also shows that SGD with dropout converges to the stationary points of $\mathcal{D}(W)$. Furthermore, we remark that the constraint to a set $\mathcal{H}$, while apparently restrictive, is in practice equivalent to assuming bounded iterates of SGD. Beyond the previous convergence guarantee, a convergence rate would yield more insight into the trade-offs of the algorithm, especially in the dependence on depth.

In our second result, we go one step beyond the convergence guarantee and compute a bound for the sample complexity of SGD with dropout to an ϵ-stationary point of a generic smooth nonconvex function $\mathsf{D}(W)$. We say $W \in \mathcal{W}$ is an ϵ-stationary point of D if $\|\nabla \mathsf{D}(W)\|_2 \leq \epsilon$ holds. Note that stationary points are not necessarily minima, but the sample complexity, understood as the number of iterations T required to reach ϵ-stationarity, is usually associated with the complexity of the function to be optimized.

For a generic smooth nonconvex function $\mathsf{D}(W)$, we consider dropout to be SGD with the update in (4), where filters F are chosen independently at each step and are $\{0,1\}$-valued for each parameter. In our result we assume boundedness and Lipschitzness conditions on $\mathsf{D}(W)$. Moreover, under some additional assumptions on the loss function, examples of NNs with sigmoid activation functions $\sigma(t) = 1/(1+\exp(-t))$ are also covered by our result. In this particular case, $\mathsf{D}(W) = \mathcal{D}(W)$ holds with the definition in (6). For the general case we prove the following:

Result 2 *(Informal statement of Proposition 7.) Assume that* $\mathsf{D}(W)$ *has enough regularity and satisfies some boundedness and Lipschitzness assumptions. Let* $W^{\{t\}}$ *be iterates of* (5). *For any* $T \in \mathbb{N}$ *there exist* $c > 0$ *and* $c_1, c_2 > 0$ *and* $\alpha^{\{t\}} = \eta$ *constant such that if* $p > c/T$, *then as* $T \to \infty$,

$$\min_{t\in[T]} \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{\{t\}})\|_2^2\Big] = O\Big(\sqrt{\frac{p(c_1+(1-p)c_2)}{T}}\Big). \tag{8}$$

Hence, at least T iterations of dropout-like SGD algorithms are required to reach an $O((p(c_1+(1-p)c_2)/T)^{1/4})$-stationary point of nonconvex smooth functions in expectation. Here, c_1, c_2 are constants depending on the data and function, respectively. Compared to the theoretical optimum rate of $O(T^{-1/4})$ for SGD on nonconvex smooth functions (Drori and Shamir, 2020), this result shows that dropout changes the optimization landscape and approximate stationary points are easier to find depending on the dropout probability. In this setting, we also consider the complexity when we scale the weights by a factor $1/p$ during training, which is commonly used to compensate the effect of dropout on the convergence rate.

It must be emphasized that Proposition 7 does not assume much structure on the objective function. As consequence, in spite of the fact that the bound in (8) holds in some settings with deep NNs, the depth of such NN would appear only *implicitly* in the constants c_1, c_2. In order to determine the dependence between the convergence rate and the depth of a NN *explicitly*, one must exploit the specific structure of a NN, which we leverage in our next result.

Our third result in this paper is an explicit upper bound for the rate of convergence of regular GD on the limiting ODEs of dropout algorithms for arborescences (a class of trees, see Figure 1c for an example), of arbitrary depth with linear activation functions $\sigma(t) = t$. In particular, we will consider the update rule

$$W^{\{t+1\}} = W^{\{t\}} - \alpha\nabla\mathcal{D}(W^{\{t\}}). \tag{9}$$

Analyzing the convergence of training algorithms on simplified NNs with linear activation functions is commonly used to gain insight into more complex models, see e.g. (Arora et al., 2019; Shamir, 2019; Bartlett et al., 2018). Even without a dropout algorithm present, this task already provides a substantial theoretical challenge as the optimization landscape is nonconvex. Our choice to restrict the analysis to arborescences allows us to quantitatively tie our upper bound for the convergence rate to the depth and the number of paths within the arborescence. We prove the following:

Result 3 *(Informal statement of Proposition 9.) Assume that the base graph G of the NN is an arborescence of depth L with $|\mathcal{L}(G)|$ leaves and the filters F follow the distribution prescribed by* Dropconnect *or* Dropout *with dropout probability* $1-p$ *(see Proposition 9). Then there exist $\alpha > 0$ and $1 > \eta > 0$ depending on the initialization such that the iterates of* (9) *satisfy*

$$\mathcal{D}(W^{\{t\}}) - \min_W \mathcal{D}(W) \leq \big(\mathcal{D}(W^{\{0\}}) - \min_W \mathcal{D}(W)\big) \exp(-\omega t/2), \tag{10}$$

with

$$\omega = \mathrm{O}\Big(\frac{p^L}{L|\mathcal{L}(G)|^2}\eta^{2L}\Big). \tag{11}$$

One important consequence of this result is that the convergence rate exponent indeed deteriorates by a factor p^L in these NNs. Finally, we complement this result with numerical experiments. We target the dependency of the convergence on p for more realistic wider and nonlinear networks on commonly used data sets. Perhaps surprisingly, we do not observe an exponential decrease of the convergence rate exponent due to dropout in these simulations. We will offer some heuristic explanation for this result by looking at the update rate of a generic weight.

Our results lead to the following consequences. First, whenever the iterates of a dropout algorithm with ℓ_2-loss are bounded, they are guaranteed to converge to a stationary point of the risk function $\mathcal{D}(W)$ induced by the dropout algorithm. Secondly, we prove rigorously that the convergence rate when training with e.g. *Dropout* or *Dropconnect* can change the convergence rate on the empirical risk depending on p and in arborescences can decrease by as much as a factor p^L. For more realistic wider networks, however, we conduct numerical experiments that suggest that the convergence rate is not necessarily affected by depth as much across different dropout rates $1-p$ in neural networks with just a few layers of dropout. As a consequence, we expect that training neural networks with many dropout layers compared to its width may result in a slow empirical risk minimization and that a small dropout rate may beneficial in these cases.

Our findings also motivate further theoretical study of the convergence rate of dropout for deep and wide networks. We namely suspect that there is a transition regime for the convergence rate of the empirical risk minimization with neural networks trained with dropout. Such transition would affect the dependence on p and would be observed when going from deep networks with small width and many layers of dropout, where dependence on the rate may depend on p exponentially in the number of dropout layers, to networks with a few layers of dropout but very wide, where dependence is not exponential anymore but polynomial in p and mostly independent of the number of dropout layers.

1.3 Literature Overview

The first description of a dropout algorithm was by Hinton et al. (2012). Diverse variants of the algorithm have appeared since, including versions in which edges are dropped (Wan et al., 2013); groups of edges are dropped from the input layer (DeVries and Taylor, 2017); the distribution of the filters are Gaussian (Kingma et al., 2015; Molchanov et al., 2017); the removal probabilities change adaptively (Ba and Frey, 2013; Li et al., 2016); and that are suitable for recurrent NNs (Zaremba et al., 2014; Semeniuta et al., 2016). The performance of the original algorithm has been investigated on data sets (Hinton et al., 2012; Srivastava et al., 2014), and dropout algorithms have found application in e.g. image classification (Krizhevsky et al., 2012), handwriting recognition (Pham et al., 2014), heart sound classification (Kay and Agarwal, 2016), and drug discovery in cancer research (Urban et al., 2018).

Theoretical studies of dropout algorithms have focused on their regularization effect. The effect was first noted by Hinton et al. (2012); Srivastava et al. (2014), and subsequently investigated in-depth for both linear NNs as well as nonlinear NNs by Baldi and Sadowski (2013); Wager et al. (2013); Baldi and Sadowski (2014); Wei et al. (2020). Within the context of matrix factorization, it has been shown that *Dropout*'s regularization induces a shrinkage and a thresholding of the singular values of the matrix at the optimum (Cavazza et al., 2018). Characterizations of *Dropout*'s risk function and *Dropout*'s regularizer for (usually linear) NNs can be found in Mianjy et al. (2018); Mianjy and Arora (2019); Pal et al. (2020). Random networks with *Dropout* have been also studied in Sicking et al. (2020) and in Huang et al. (2019).

Detailed theoretical investigations into the convergence of dropout algorithms are however relatively scarce. While revising this paper, new results appeared and these now give insight into the convergence rate of *Dropout* in ReLU shallow NNs for a classification task (Mianjy and Arora, 2020). In Mianjy and Arora (2020), it is shown that $\mathrm{O}(1/\epsilon)$ iterations of SGD to reach ϵ-suboptimality for the test error are required; interestingly, it is independent of the dropout probability because of their assumption that the data distribution is separable by a margin in a particular Reproducing Kernel Hilbert space. Compared to our generic convergence result, we do not assume structure on the predictor or data and look instead at the iterations required to reach ϵ-stationarity in nonconvex functions using dropout-like SGD. A study of the asymptotic convergence rate of *Dropout* and *Dropconnect* on shallow linear neural networks has also appeared recently (Senen-Cerda and Sanders, 2022). There, an asymptotic convergence rate for dropout linear shallow networks is provided. Namely, for wide linear shallow networks with width D and dropout probability $1 - p > 0$ a local convergence rate close to a minimum of $O(p(1-p)/(pD+1-p))$ is found. Finally, it must be noted that convergence properties have been thoroughly studied within the context of NNs being trained without dropout algorithms, see e.g. Arora et al. (2019); Shamir (2019); Zou et al. (2020); Gao et al. (2021) and references therein.

Dropout algorithms can, by construction, be understood as forms of SGD. More generally, dropout algorithms are all stochastic approximation algorithms. The first stochastic approximations algorithms were introduced by Robbins and Monro (1951); Kiefer and Wolfowitz (1952), and have been subject to enormous literature due to their ubiquity. For overviews and their application to NNs, we refer to books by Kushner and Yin (2003); Borkar (2009); Bertsekas and Tsitsiklis (1995).

Notation. In this paper we index deterministic sequences with curly brackets: $\alpha^{\{1\}}, \beta^{\{1\}}$, etc. This distinguishes them from sequences of random variables, which we index using square brackets, e.g. $X^{[1]}, Y^{[1]}$, etc.

Deterministic vectors are written in lower case like $x \in \mathbb{R}^d$, but an exception is made for random variables (which are always capitalized). Matrices are also always capitalized. For a function $\sigma : \mathbb{R} \to \mathbb{R}$ and a matrix $A \in \mathbb{R}^{a\times b}$, $a, b \geq 1$, we denote by $\sigma(A)$ the matrix with σ applied componentwise to A. Subscripts will be used to denote the entries of any tensor, e.g. x_i, $A_{i,j}$, or $T_{i,j,l}$. For any vector $x \in \mathbb{R}^d$, the ℓ_2-norm is defined as $\|x\|_2 \triangleq (\sum_{i=1}^{d} |x_i|^2)^{1/2}$. For any matrix $A \in \mathbb{R}^{a\times b}$, the Frobenius norm is defined as $\|A\|_{\mathrm{F}} \triangleq (\sum_{i=1}^{a} \sum_{j=1}^{b} |A_{i,j}|^2)^{1/2}$. For two matrices A, B, the Hadamard (componentwise) product is denoted by $A \odot B$.

Let $\mathbb{N}_+$ be the strictly positive integers and $\mathbb{N}_0 \triangleq \mathbb{N}_+ \cup \{0\}$. For $l \in \mathbb{N}_+$, we denote $[l] = \{1, \dots, l\}$. For a function $g \in C^2(\mathbb{R}^n)$, we denote the gradient and Hessian of g with respect to the Euclidean norm $\|\cdot\|_2$ in $\mathbb{R}^n$ by ∇g and $\nabla^2 g$, respectively.

2. Model

We now formally define NNs, which we had depicted in Figure 1, as well as the class of activation functions that we will use for the convergence guarantee in our first result below.

2.1 Neural Networks, and their Structure

Let L denote the number of layers in the NN, and $d_l \in \mathbb{N}_+$ the output dimension of layer $l = 1, \dots, L$. Let $W_{l+1} \in \mathbb{R}^{d_{l+1}\times d_l}$ denote the matrix of weights in between layers l and $l+1$ for $l = 0, 1, \dots, L-1$. Denote $W = (W_L, \dots, W_1) \in \mathcal{W}$ with $\mathcal{W} \triangleq \mathbb{R}^{d_L \times d_{L-1}} \times \cdots \times \mathbb{R}^{d_1\times d_0}$ the set of all possible weights. In this paper, we consider NNs without biases.

Definition 4 *Let σ be an activation function $\sigma : \mathbb{R} \to \mathbb{R}$. A* Neural Network (NN) *with* L layers *is given by the class of functions $\Psi_W : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ defined iteratively by*

$$A_0 = x, \quad A_i = \sigma(W_i A_{i-1}) \quad \forall i \in \{1, \dots, L-2\}, \quad \Psi_W(x) = W_L A_{L-1} = A_L. \tag{12}$$

Canonical activation functions include the Rectified Linear Unit (ReLU) function $\sigma(t) = \max\{0, t\}$, the sigmoid function $\sigma(t) = 1/(1 + \mathrm{e}^{-t})$, and the linear function $\sigma(t) = t$. In Sections 2 and 3 we restrict to the case that σ belongs to a class of polynomially bounded differentiable functions.

Definition 5 *For $\sigma : \mathbb{R} \to \mathbb{R}$ differentiable, denote the lth derivative of σ by $\sigma^{(l)}$. The set of polynomially bounded maps with continuous derivatives up to order $r \in \mathbb{N}_0$ is given by*

$$C^r_{\mathrm{PB}}(\mathbb{R}) = \big\{\sigma \in C^r(\mathbb{R}) \big| \forall l = 0, \ldots, r \ \exists k_l > 0 : \sup_{x\in\mathbb{R}} |\sigma^{(l)}(x)(1+x^2)^{-k_l}| < \infty \big\}.$$

Note that the linear and sigmoid activation function both belong to $C^r_{PB}(\mathbb{R})$ for any $r \in \mathbb{N}_0$. Also, any polynomial activation function $P(x) \in \mathbb{R}[x]$ belongs to $C^{\deg(P)}_{\mathrm{PB}}(\mathbb{R})$. The ReLU activation function is not in $C^r_{\mathrm{PB}}(\mathbb{R})$ for any $r \in \mathbb{N}_0$. However, because the class $C^r_{\mathrm{PB}}(\mathbb{R})$ contains polynomials of any degree, we can approximate cases such as ReLU by using, e.g., the softplus activation function $\sigma_t(x) = \log(1 + \exp(tx))/t$, which satisfies that $\lim_{t\to\infty} \sigma_t(x) = \mathrm{ReLU}(x)$ for every $x \in \mathbb{R}$. Note that the softplus activation function belongs to $C^2_{\mathrm{PB}}(\mathbb{R})$.

2.2 Backpropagation, and SGD

In Section 1.1 we have defined the risk $\mathcal{U}(W)$ that in the previous notation now depends on a loss $l : \mathbb{R}^{d_L} \times \mathbb{R}^{d_L} \to \mathbb{R}$. Throughout this article, we will specify the Euclidean ℓ_2-norm $l(x, y) \triangleq \|x - y\|_2^2$ as our loss function of interest without loss of generality. [1]

Furthermore, in the definition of $\mathcal{U}(W)$ in (1), we make no distinction between an oracle risk function or empirical risk function. Both situations are covered by the definition in (1). Hence, our results cover the empirical risk case when we have a finite number of samples, as well as the online learning case, where a new sample is provided at each step of SGD. What we do assume is that one has the ability to repeatedly draw independent and identically distributed samples either distribution.

In an attempt to find a critical point in the set $\arg\min_W \mathcal{U}(W)$, as mentioned in (1.1), SGD is commonly used. Let $\{(Y^{[t]}, X^{[t]})\}_{t\in\mathbb{N}_+}$ be a sequence of independent copies of (X, Y), let $W^{[0]} \in \mathcal{W}$ be an arbitrary nonrandom initialization of the weights. For $i = 1, \ldots, L$, $r = 1, \ldots, d_{i+1}$, $l = 1, \ldots, d_i$, the weights are iteratively updated according to

$$W^{[t+1]}_{i,r,l} = W^{[t]}_{i,r,l} - \alpha^{\{t+1\}} \big(\mathrm{B}_{W^{[t]}}(X^{[t+1]}, Y^{[t+1]})\big)_{i,r,l} \tag{13}$$

for $t = 0, 1, 2,$ *et cetera*. Here $\{\alpha^{\{t\}}\}_{t\in\mathbb{N}_+}$ denotes a positive, deterministic step size sequence, and the estimate of the gradient $B_W(\cdot,\cdot) = \nabla_W l(\Psi_W(\cdot), \cdot)$ is computed using the backpropagation algorithm, which is given in Definition 15 in Appendix A. The stochastic gradient is an unbiased estimate of the gradient of $\mathcal{U}(W)$. In particular, we have

$$\mathbb{E}[\big(\mathrm{B}_W(X, Y)\big)_{i,r,l}] = \mathbb{E}\Big[\frac{\partial l(\Psi_W(x), y)}{\partial W_{i,r,l}}\Big] = \frac{\partial \mathcal{U}(W)}{\partial W_{i,r,l}} = (\nabla U)_{i,r,l}. \tag{14}$$

2.3 Dropout Algorithms, and their Risk Functions

Dropout algorithms use $\{0, 1\}$-valued random matrices as filters of weights during the backpropagation step of SGD. More precisely, we examine the following class of dropout algorithms. Let $(F, X, Y) : \Omega \to \{0, 1\}^{d_L \times d_{L-1}} \times \ldots \times \{0, 1\}^{d_1\times d_0} \times \mathbb{R}^{d_0} \times \mathbb{R}^{d_L}$ be a random variable

[1] The results can be extended to other smooth loss functions $l(x, y)$ whose partial derivatives can be bounded by polynomials of finite degree.

on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Here, we write $F = (F_L, \ldots, F_1)$ and $F_{i+1} \in \{0,1\}^{d_{i+1}\times d_i}$ for $i = 0, \ldots, L-1$, similar to how we notate weight matrices. Let $\{(F^{[t]}, X^{[t]}, Y^{[t]})\}_{t\in\mathbb{N}_+}$ be a sequence of independent copies of (F, X, Y). In tensor notation, the weights are updated by using (2) with the random direction $\Delta^{[t+1]}$ for dropout given in (4). For each dropout algorithm a different filter distribution will be chosen. We can mention a few:

(i) In canonical *Dropout* (Hinton et al., 2012), $F_{i,r,l'} = F_{i,r,l} \sim \text{Bernoulli}(p)$ for any $l, l' \in [d_i]$ with $p = 1/2$.

(ii) In *Dropconnect* (Wan et al., 2013), $F_{i,r,l} \sim \text{Bernoulli}(p)$ for all i, r, l with $p = 1/2$.

(iii) In *Cutout* (DeVries and Taylor, 2017), $F_{1,r,l} = 0$ whenever $|r - S_1| < c$, $c \in \mathbb{N}_+$ and $|l - S_2| < c$ with $(S_1, S_2) \sim \text{Uniform}([d_1] \times [d_0])$.

In fact, the class of dropout algorithms we consider is quite large. For example, $F^{[t]}$ can depend on $(X^{[t]}, Y^{[t]})$, and $F_i^{[t]}$ does not need to have the same distribution as $F_j^{[t]}$ for $i \neq j$. Recall, however, that if for some filter $F_{i,r,l}^{[t+1]} = 0$ for some i, r, l, then in (2) , $\Delta_{i,r,l}^{[t]} = 0$ and we have $W_{i,r,l}^{[t]} = W_{i,r,l}^{[t+1]}$. In other words, filtered variables are not updated with these dropout algorithms.

If $F^{[t]}$ is independent of $(X^{[t]}, Y^{[t]})$ for each $t \in \mathbb{N}_0$ and Ω countable, then the dropout algorithm's risk function in (6) simplifies to

$$\mathcal{D}(W) = \sum_f \mathbb{P}[F = f] \sum_{x,y} l(\Psi_{f\odot W}(x), y)\mathbb{P}[(X,Y) = (x,y)]. \tag{15}$$

Here the sums are over all possible outcomes of the random variables F and (X, Y), respectively. One implication of Proposition 6 in the result of the next Section 3 is that dropout algorithms of the kind in (2), (4) converge to a critical point of (6).

3. Convergence of Projected Dropout Algorithms

Our first result pertains to the convergence of dropout algorithms for a wide range of activation functions and dropout filters. While convergence is expected in practice, we prove such convergence rigorously. In order to control the iterates of the stochastic algorithm, we project the iterates into a compact set. The projection assumption is common when investigating the convergence of stochastic algorithms (Kushner and Yin, 2003; Borkar, 2009; Bertsekas and Tsitsiklis, 1995; Oymak, 2018); it essentially bounds the weights. For example, for $V^{[t]} \in \mathbb{R}$ and an update function $f : \mathbb{R} \to \mathbb{R}$, $f(V^{[t]})$ is projected onto an interval $[a, b]$ is by clipping and setting $V^{[t+1]} = \min\{\max\{f(V^{[t]}), a\}, b\}$. There are also results involving generalization bounds for NNs where bounded weights play a role in controlling the learning capacity of the NN (Neyshabur et al., 2015).

3.1 Almost Sure Convergence

We first consider the notation and assumptions regarding the projection step of SGD. Let $\mathcal{H} \subseteq \mathcal{W}$ be a convex compact nonempty set and let $\mathrm{P}_{\mathcal{H}} : \mathcal{W} \to \mathcal{H}$ be the projection onto $\mathcal{H}$. By compactness and convexity of $\mathcal{H}$, the projection is unique. In a projected dropout algorithm, the weight update in (2) is replaced by (5). Because of the projection, our analysis will tie the limiting behavior of (5) to a *projected* ODE. To state such type of ODE, we need to define a *constraint term* $\pi(W)$, which is defined as the minimum vector required to keep the solution of the gradient flow

$$\frac{\mathrm{d}W}{\mathrm{d}t} = -\nabla_W \mathcal{D}|_{\mathcal{H}}(W) + \pi(W) \tag{16}$$

in $\mathcal{H}$. Appendix C defines the projection term carefully for the case that $\mathcal{H}$'s boundary is piecewise smooth. While this projection step is used to show the results in full generality, in practice it is rarely used, and we do not use projection in Sections 3.2–4 or the numerical experiments of Section 5.1. Finally, define the set of stationary points

$$S_{\mathcal{H}} \triangleq \{W \in \mathcal{H} \ : \ -\nabla_W \mathcal{D}|_{\mathcal{H}}(W) + \pi(W) = 0\}. \tag{17}$$

The set $S_{\mathcal{H}}$ can be divided into a countable number of disjoint compact and connected subsets $S_1, S_2, \cdots$, say. We choose the following set of assumptions:

(N1) $\sigma \in C^2_{\mathrm{PB}}(\mathbb{R})$.
(N2) $\mathbb{E}[\|Y\|_2^m \|X\|_2^n] < \infty \, \forall m \in \{0,1,2\}, n \in \mathbb{N}_0$.
(N3) The random variables $(F^{[t]}; X^{[t]}; Y^{[t]})_{t\in\mathbb{N}}$ are independent copies of (F, X, Y).
(N4) The step sizes $\alpha^{\{t\}}$ satisfy

$$\sum_{t=1}^{\infty} \alpha^{\{t\}} = \infty, \quad \sum_{t=1}^{\infty} (\alpha^{\{t\}})^2 < \infty. \tag{18}$$

(N5) $\sigma \in C^r_{\mathrm{PB}}(\mathbb{R})$, with $\dim(\mathcal{W}) \leq r$.
(N6) $-\nabla_W \mathcal{D}|_{\mathcal{H}}(W) + \pi(W) \neq 0$ whenever $\nabla_W \mathcal{D}|_{\mathcal{H}}(W) \neq 0$.

We are now in position to state our first result:

Proposition 6 *Let* $\{W^{[t]}\}_{t\in\mathbb{N}_0}$ *be the sequence of random variables generated by* (5) *with* (4) *on a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$. *Under assumptions (N1)–(N4) , there is a set* $N \subset \Omega$ *of probability zero such that for* $\omega \notin N$, $\{W^{[t]}(\omega)\}$ *converges to a limit set of the projected ODE in* (16). *If moreover (N5)–(N6) hold, then for almost all* $\omega \in \Omega$, $\{W^{[t]}(\omega)\}_{t\in\mathbb{N}}$ *converges to a unique point in* $\{W \in \mathcal{H} | \nabla\mathcal{D}|_{\mathcal{H}}(W) = 0\}$.

Theoretically, Proposition 6 guarantees that projected dropout algorithms converge for regression with the ℓ_2-norm almost surely. Proposition 6 implies that if one is using a regular *nonprojected* dropout algorithm and one sees that the iterates $\{W^{[t]}\}_{t>0}$ are bounded, then these iterates are in fact converging to a stationary point of (6). Assumptions (N5)–(N6) are technical but are expected to hold in many cases. In particular, (N5) holds for the uniformly convergent approximation to a ReLU activation function given by softplus $\sigma_t(x) = \log(1 + \exp(tx))/t$, and holds for many smooth activation functions. Also (N6) is expected to hold when $\mathcal{H}$ is generic polytope for which the gradient $\nabla\mathcal{D}$ is not exactly orthogonal to the normal to the surface.

Observe also that Proposition 6 holds remarkably generally. For example, the dependence structure of (F, X, Y) as random variables is not restricted; it covers commonly used dropout algorithms such as *Dropout*, *Dropconnect*, and *Cutout*; and it holds for differentiable activation functions. Proposition 6 includes also online and offline learning, depending on the distribution μ from which we sample.

Our proof of Proposition 6 is in Appendix D and relies on the framework of stochastic approximation in (Kushner and Yin, 2003, Theorem 2.1, p. 127). In the background the stochastic process $\{W^{[t]}\}_{t>0}$ is being scaled in both parameter space and time so that the resulting sample paths provably converge to the gradient flow in (16). Examining the proof, we expect that Proposition 6 can be extended to cases where the filters as random variables have finite moments, for example, when they are Gaussian distributed (Molchanov et al., 2017). Concretely, the proofs of Lemmas 17 and 18 in Appendix D rely only on the assumption that F has finite moments, and may therefore be extended.

3.2 Generic Sample Complexity for Dropout SGD

Examining Proposition 6, we note that it does not give insight into the convergence rate or the precise stationary point of $\mathcal{D}(W)$ to which the iterates $\{W^{[t]}\}$ converge. A related goal in stochastic optimization is to ask for the number of iterations of (2) required to achieve a point close to stationarity in expectation, also referred to the sample complexity of the algorithm. We say $W \in \mathcal{W}$ is an ϵ-stationary point of a differentiable function D if $\|\nabla \mathsf{D}(W)\|_2 \leq \epsilon$ holds. For nonconvex functions D with a Lipschitz continuous gradient $\nabla \mathsf{D}$, SGD convergence to an ϵ-stationary point in expectation can be achieved in $O(\epsilon^{-4})$ iterations; see Bottou et al. (2018); Drori and Shamir (2020).

We will consider nonconvex functions with a Lipschitz continuous gradient and assume that the filters F and the data $Z = (X, Y)$ are independent. We will also assume that the distribution of Z is well-behaved so as to guarantee that we also have the following relations for the functions r, U and D:

$$\begin{aligned} \mathsf{U}(W) &= \mathbb{E}_Z[r(W, Z)], \\ \mathsf{D}(W) &= \mathbb{E}_F[\mathsf{U}(F \odot W)], \quad \text{and} \\ \nabla \mathsf{D}(W) &= \mathbb{E}_F[F \odot \nabla \mathsf{U}(F \odot W)] = \mathbb{E}_{F,Z}[F \odot \nabla r(F \odot W, Z)]. \end{aligned} \tag{19}$$

Note that the function r in this setting includes the loss function formulation from (1) with

$$r(W, Z) = l(\Psi_W(X), Y), \quad \text{and} \quad Z = (X, Y), \tag{20}$$

and in general, at time t the update rule will be

$$W^{[t+1]} = W^{[t]} - \alpha^{\{t+1\}} F^{[t]} \odot \nabla r\Big(W^{[t]} \odot F^{[t]}, Z^{[t]}\Big). \tag{21}$$

In the case of *dropout*, for example, we expect that the sample complexity of finding an ϵ-stationary point for the empirical risk will change depending on the dropout probability $1-p$. In particular, if $p \downarrow 0$ and $\|\nabla \mathsf{U}(W)\|_\infty < C$ holds for any $W \in \mathcal{W}$, then $\nabla \mathsf{D}(W) = \mathbb{E}_F[F \odot \nabla \mathsf{U}(F \odot W)] = O(pC)$. On the other hand if $p \uparrow 1$, then the variance of $F \odot \nabla \mathsf{U}(F \odot W)$, will also be small. We make these intuitions rigorous in the next proposition. For some $N \in \mathbb{N}$, we let $\mathcal{W} = \mathbb{R}^N$ be the parameter space and $z \in \mathcal{Z} \subseteq \mathbb{R}^d$ a Lebesgue measurable set. We assume the following:

- (Q1) $r \in C^1(\mathcal{W}, \mathcal{Z})$ and $\sup_{W \in \mathcal{W}, Z \in \mathcal{Z}} \; |r(W, Z)| < M$.
- (Q2) $\sup_{W \in \mathcal{W}, Z \in \mathcal{Z}} \; \|\nabla r(W, Z)\|_2 < S$.
- (Q3) $\nabla \mathsf{U}(W)$ is Lipschitz with Lipschitz constant ℓ (also referred to as U being ℓ-smooth).
- (Q4) The random variable $F : \Omega \to \{0,1\}^N$ satisfies $\mathbb{E}[F] = p(1, \ldots, 1) \in \mathcal{W}$ for $p \in (0, 1]$.
- (Q5) The iterates $(W^{[t]})_t$ of (2) are bounded, that is, $\sup_t \|W^{[t]}\|_2 < R$ almost surely.

Except for (Q4) and (Q5), all other assumptions are routinely used in sample complexity analysis. While the assumptions of Proposition 7 below hold for general nonconvex smooth functions D, in the case of NNs and the setting in (20) we remark that there are examples that satisfy these assumptions such as the following one:

Example 1 *In a binary classification setting, the set $\mathcal{Z}$ is compact, that is, the data pairs $(x, y) \in \mathcal{Z}$ take values in a compact set where $y \in \{0, 1\}$ are labels for the two classes. A NN, denoted by $\tilde{\Psi}_W(\cdot)$, uses sigmoid activation functions $\sigma(t) = 1/1 + \exp(-t)$ with output in $\mathbb{R}$. The output of $\tilde{\Psi}_W$ is then used for binary classification with a logistic map, that is, the predicted probability of belonging to one of the classes is given by $\Psi_W(x) = 1/(1 + \exp(-\tilde{\Psi}_W(x))$. In this setting, assumptions (Q1)–(Q3) will hold if the loss l is also smooth (such as the ℓ_2-loss). In this case, we have $\mathcal{D}(W) = \mathsf{D}(W)$ and the constants in (Q1)–(Q5) will also indirectly depend on the depth and width of the NN.*

Regarding (Q4), note that it allows for dependencies between filters. We also assume (Q5) for the sake of simplicity: we could instead use projected SGD with updates from (5) instead of (Q5), but using projected SGD would leave the scalings in p and T invariant. [2] Recall that $\mathsf{D}(W) = \mathbb{E}_F[\mathsf{U}(F \odot W)]$. The proof the following proposition can be found in Appendix E.

Proposition 7 *Let $(F^{[t]})_{t \in \mathbb{N}}$ be a sequence of independent random variables with distribution F. Let $W^{[t]}$ be iterates of* (21). *Assume (Q1)–(Q5). Define $J = S^2 + \frac{3}{2} N^2 (\ell^2 R^2 + 2\ell R)$.*
(a) *Let $T \in \mathbb{N}_+$. If $p > M\ell/(NS^2T)$, then there exists a constant stepsize $\alpha^{\{t\}} = \eta > 0$ such that for all $t \in [T]$,*

$$\min_{t \in [T]} \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]})\|_2^2\Big] \le 4\sqrt{p(S^2 + (1-p)J)}\sqrt{\frac{M\ell N}{T}}. \tag{22}$$

(b) *Let $T \geq 4$. There exists a sequence of decreasing stepsizes satisfying $\alpha^{\{t\}} = 1/(\ell\sqrt{t})$ for all $t \in [T]$ such that*

$$\min_{t \in [T]} \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]})\|_2^2\Big] \le \frac{4M\ell^2 + 4Np(S^2 + (1-p)J)\log(T)}{\sqrt{T}}. \tag{23}$$

[2] With projected SGD, we would moreover have to use the expression $\nabla \mathsf{U}^p(w) = (w - \mathrm{P}_{\mathcal{H}}(W^{[t]} - \alpha^{\{t+1\}}\Delta^{[t+1]}))/\alpha^{\{t+1\}}$, which makes the analysis more tedious. Note that $\nabla \mathsf{U}^p(w) = \nabla \mathsf{U}(w)$ whenever $w \in \mathrm{int}(\mathcal{H})$. See Bubeck et al. (2015) for an example of such analysis.

In Proposition 7, we observe that finding approximate stationary points is easier with a larger dropout probability $1-p$ for a wide range of filter distributions like those determining *dropout* and *dropconnect*, as guaranteed by (Q4). In Proposition 7(a) we also see a dependence of the convergence rate on $\sqrt{p(S^2+(1-p)J)}$. The term pS^2 corresponds to the variance of the gradient due the distribution of data in $\mathcal{Z}$ and decreases with p; while the term $p(1-p)J$ stems from the variance due to dropout. Note that the sum achieves a maximum for $p \in (0,1)$. We note that Proposition 7 does not suggest that the convergence to minima is faster for smaller p. In particular, saddle points can become easier to find as $p \uparrow 0$. As seen later in the numerical experiments with NNs in Section 5.1, or in similar work from Mianjy and Arora (2020); Senen-Cerda and Sanders (2022), the NN structure and data distribution can change the convergence rate dependence on the dropout probability. As an example, in Senen-Cerda and Sanders (2022) it is suggested that the convergence rate dependence on p and the width of the NN can have different regimes depending on whether we are close to a minimum or not. Similarly, smaller p does not necessarily improve generalization. In particular, if the dropout probability $1-p$ is large, the optimization landscape will be flat with many approximate stationary points. In this case, SGD with dropout with a limited sample complexity of T iterations will not explore the landscape as much as when using a smaller dropout probability. With a flatter landscape in mind, it may be better in the complexity trade-off to use a larger p for finding an approximate minimum and generalize better instead of finding a stationary point.

A possible approach to avoid the flattening of the landscape is to scale the weights appropriately during training. This is, for example, what is conducted in practice in some implementations of dropout.[3] Assuming (Q4) holds, we consider the update rule

$$W^{[t+1]} = W^{[t]} - \alpha^{\{t+1\}} \frac{F^{[t]}}{p} \odot \nabla r\Big(W^{[t]} \odot \frac{F^{[t]}}{p}, Z^{[t]}\Big). \tag{24}$$

With (24), the use of filters is compensated by increasing the size of the updates and weights accordingly. In this case, SGD with this update rule is actually minimizing the function

$$\tilde{\mathsf{D}}(W) = \mathsf{D}\Big(\frac{W}{p}\Big), \tag{25}$$

which also compensates in expectation the effect of the filters. With the update rule in (24), we can again obtain an expression for the complexity of finding an ϵ-stationary point of $\tilde{\mathsf{D}}(W)$. The following is proved in Appendix E:

Proposition 8 *Let $(F^{[t]})_{t\in\mathbb{N}}$ be a sequence of independent random variables with distribution F. Assume (Q1)–(Q5). Let $W^{[t]}$ be iterates of* (24). *Let $T \in \mathbb{N}_+$. If $p > M\ell/(NS^2T)$, then there exists a constant stepsize $\alpha^{\{t\}} = \eta > 0$ such that for all $t \in [T]$,*

$$\min_{t\in[T]} \mathbb{E}\Big[\|\nabla\tilde{\mathsf{D}}(W^{[t]})\|_2^2\Big] \le 4\sqrt{\frac{1}{p^3}\Big(S^2 + \frac{(1-p)}{p^2}\Big(p^2S^2 + \frac{3}{2}N^2\big(p\ell^2R^2 + 2\ell R\big)\Big)\Big)}\sqrt{\frac{M\ell N}{T}}. \tag{26}$$

[3] For example, scaling is implemented with the Dropout layer implementation in *Keras*, `https://keras.io/`.

Proposition 8 shows that for the scaled dropout SGD of (24) the complexity of finding an ϵ-stationary point monotonically increases with $1-p$. This result contrasts with Proposition 7, where a different behavior was observed. We remark, however, that this result assumes (Q5), which for small p cannot realistically hold since a bound R for the norm of the weights may also scale by a factor $1/p$. This result, just like with Proposition 7, also does not imply that good weights $W \in \mathcal{W}$ become easier to find by using the update (24). Indeed, scaling partially avoids the flattening of the landscape—the Lipschitz constant of $\nabla\tilde{\mathsf{D}}$ is namely scaled by a factor $1/p^2$—but the variance of SGD due to dropout is also increased considerably. This variance becomes dominant when the dropout rate $1-p \uparrow 1$ due to the inverse dependence on p in the sample complexity.

Propositions 7 and 8 show that the complexity of finding ϵ-stationary points heavily depends on the algorithm used. However, when we restrict the results to deep NNs such as with Example 1, the bounds do not provide much information on the dependence of the convergence rate on the depth of the network. This fact also shows the limitations of using a generic sample complexity analysis.

In order to obtain an explicit convergence rate depending on the depth, we need to use the additional structure of the NN. In the next section we will be able to compute the convergence rate to a global minimum for NNs that are shaped like arborescences and obtain an explicit bound that depends on the depth of the arborescence and the dropout probability.

4. Convergence Rate of GD on $\mathcal{D}(W)$ for Arborescences with Linear Activation

We obtained a convergence guarantee as well as a bound for the sample complexity of dropout in the previous section. Next, we focus on the convergence rate of dropout in functions that model the structure of NNs. In particular, we will derive an explicit convergence rate for dropout algorithms in the case that we have linear activations $\sigma(z) = z$ and that the NN is structured as an arborescence: see Figure 1c. Specifically, we will study the following regular GD algorithm on dropout's risk function:

$$W^{\{t+1\}} = W^{\{t\}} - \alpha \nabla \mathcal{D}(W^{\{t\}}) \quad \text{for} \quad t \in \mathbb{N}_0. \tag{27}$$

Here, we keep the step size $\alpha > 0$ fixed. Note that this algorithm generates a deterministic sequence $\{W^{\{t\}}\}_{t\in\mathbb{N}_0}$ as opposed to a sequence of random variables $\{W^{[t]}\}_{t\in\mathbb{N}_0}$ as generated by (2) or (4). We will use a linear activation function $\sigma(t) = t$, which combined with the arborescence structure will allow us to obtain an explicit convergence rate. While the iterates of (27) are not stochastic, analogous to Proposition 6, the stochastic iterates will converge to a gradient flow of an ODE, whose discretization is given in (27). Analyzing ODEs related to NNs is common in literature Tarmoun et al. (2021); Jacot et al. (2018). For more discussion on the relationship between the iterates of (27) and dropout we refer to Appendix B.

Our main convergence result in Proposition 13 below holds for general distribution functions. However, we show here the cases of *Dropout* and *Dropconnect*, which are most insightful. We use the following notation adapted from graph theory. Consider a fixed, directed *base graph* $G = (\mathcal{E}, \mathcal{V})$ without cycles in which all paths have length L, which describes a NN's structure as follows. Each vertex $v \in \mathcal{V}$ represents a neuron of the NN, and each directed edge $e = (u, v) \in \mathcal{E}$ indicates that neuron u's output is input to neuron v. Note that to each edge $e \in \mathcal{E}$ in the NN, a weight $W_e \in \mathbb{R}$ and a filter variable $F_e \in \{0, 1\}$ are associated. We will write $\mathcal{W} = \mathbb{R}^{|\mathcal{E}|}$ for simplicity. For an arborescence G, we denote by $\mathcal{L}(G)$ the edge set of leaves. Let $M > 2\delta > 0$ be real numbers and suppose that we initialize the weights $\{W_e\}_{e\in\mathcal{E}}$ as follows:

$$\begin{aligned} &M > W_e^{\{0\}} > \sqrt{2}\delta \text{ for } e \in \mathcal{E}\backslash\mathcal{L}(G) \\ &|W_l| \leq \delta/\sqrt{|\mathcal{L}(G)|} \text{ for } l \in \mathcal{L}(G). \end{aligned} \tag{28}$$

The proof of Proposition 9 is deferred to Appendix I, which is a consequence of our more general result in Proposition 13.

Proposition 9 *Assume that the base graph G is an arborescence of depth L with $|\mathcal{L}(G)|$ leaves, the activation function $\sigma(t) = t$ is linear, F is independent of (X, Y), and $\{W_e^{\{0\}}\}_{e\in\mathcal{E}}$ is initialized according to* (28)*. If the $\{F_e\}_{e\in\mathcal{E}}$ follow the distribution prescribed by* Dropconnect *or* Dropout*, then there exists $\alpha > 0$ such that the iterates of* (27) *satisfy*

$$\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\mathrm{opt}}) \leq \big(\mathcal{D}(W^{\{0\}}) - \mathcal{D}(W^{\mathrm{opt}})\big) \exp(-\omega t/2). \tag{29}$$

with

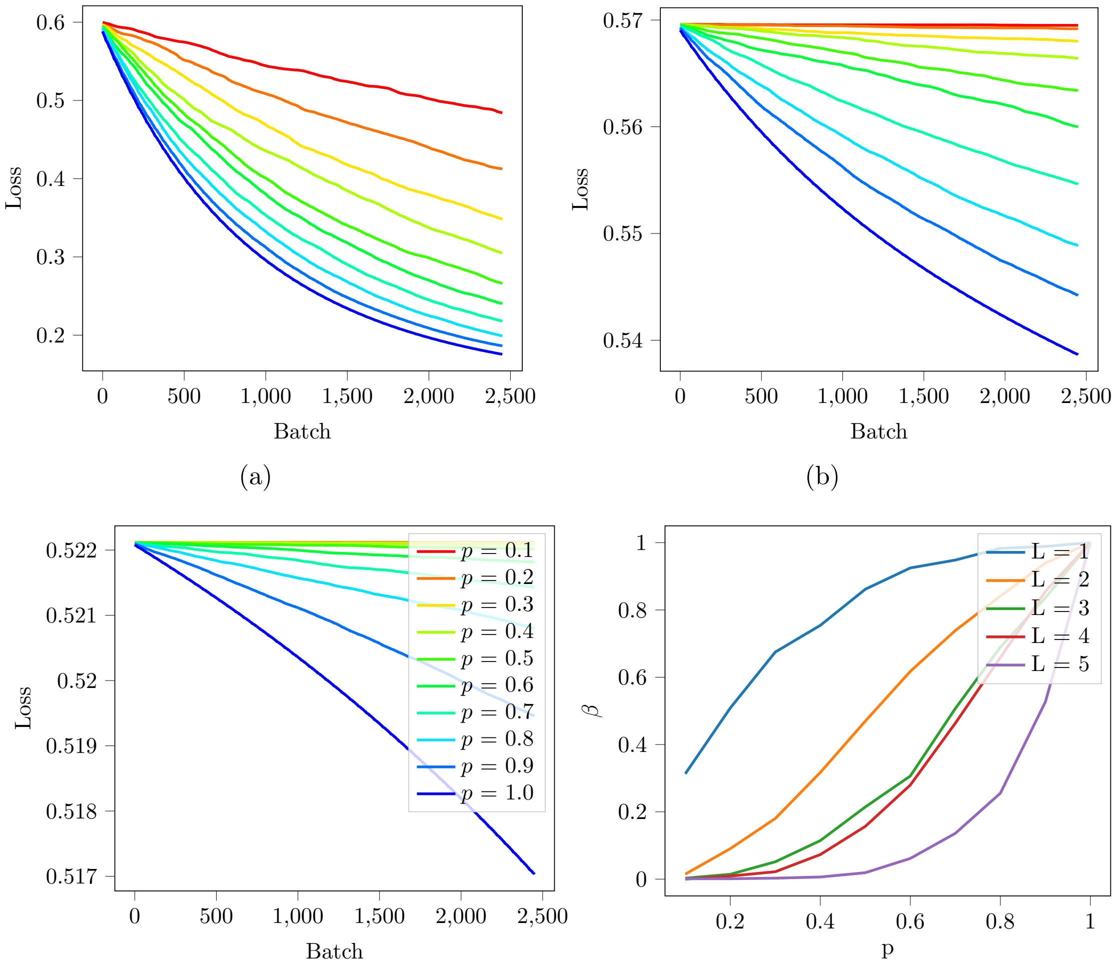
$$\omega = \mathrm{O}\Big(\frac{p^L}{L|\mathcal{L}(G)|^2}\Big(\frac{2\delta^2}{M^2}\Big)^{2L}\Big). \tag{30}$$

4.1 Discussion

In Proposition 9 we consider the cases of *Dropout* and *Dropconnect*, in which nodes or edges are dropped with probability $1 - p$, respectively. Observe that the convergence rate exponent depends on p^L and $(2\delta^2/M^2)^{2L}$ where $2\delta^2/M^2 < 1$; see (28). The first term in particular indicates that as the NN becomes deeper, the convergence rate exponent of GD with *Dropout* or *Dropconnect* will decrease by a factor p^L. The second term $(2\delta^2/M^2)^{2L}$ shows the increased difficulty of training deeper NNs and has been observed e.g., by Shamir (2019); Arora et al. (2019). The exponential dependence in L is moreover tight when using GD and is intrinsic to the method (Shamir, 2019). Hence, dropout adds another exponential dependence to the convergence rate in arborescences, which is due to the stochastic nature of the algorithm. In Figure 2 an experiment confirming this intuition on the convergence rate of dropout on a single path for different depths can be seen.

Finally, our proofs of Proposition 9 and the related more general result in Proposition 13 below can be found in Appendix H. The proof strategy is to show that a Polyak–Łojasiewicz (PL) inequality holds, which allows one to obtain convergence rates for GD on nonconvex functions (Karimi et al., 2016). The new part of the argument is that we use conserved quantities and a double induction to identify a compact set in which the iterates remain and simultaneously a PL inequality holds. The method that we develop and which is sketched in the next subsection depends intricately on the arborescence structure and cannot be readily applied to other cases.

To compare this result with more realistic models, we will examine the convergence rate of dropout in deep and wide NNs in Section 5 with a heuristic and experimental approach.



(a)

(b)

(c)

(d)

Figure 2: The average loss depending on the number of steps of SGD with dropout of the function $f(w) = (y - \prod_{i=1}^{L} w_i x)^2$ and its average convergence slope. *(a)* The average loss for $L = 1$. *(b)* The average loss for $L = 3$. *(c)* The average loss for $L = 5$. As the number of dropout layers increases, the negative effect of dropout in the convergence rate increases. Namely, we can see the loss profile corresponding to low values of p (large dropout probability) becoming flatter as we increase the number of dropout layers. *(d)* The slope β of the fit of $y = -\beta x + \gamma$ for the curves in (a), (b) and (c). The slopes β for a given l have been normalized at $p = 1$ for comparison across depths L. β can be understood to approximate the average gradient of the loss during the runtime, that is $\beta \simeq \overline{\nabla L}$. Note that as the number of dropout layers L increases, the effect of dropout on the convergence rate becomes also more pronounced, that is, for the same dropout probability the convergence rate will be slower when more dropout layers are used. This is in agreement with the conclusion in Section 4, where we expect a convergence rate depending on p^L. In this case, other effects of depth are also observed, such as a dependence on the initialization.

4.2 Sketch of the Proof

Besides the previous notation, we need to introduce notation corresponding to subgraphs and paths. Let $\mathcal{G}$ be the set of all subgraphs of the base layered directed graph G with d vertices, and let $\mathcal{E}(g)$ be the set of edges of a subgraph $g \in \mathcal{G}$. Let $\Gamma_i^j(g; e)$ be defined as the set of all paths in the directed graph g that start at vertex i, traverse edge e, and end at vertex j. If the origin or end vertices are in the input or output layer, the subscript or superscript is dropped from the notation, respectively. For every path $\gamma \triangleq (\gamma_1, \dots, \gamma_L) \in \Gamma(g)$, we write $P_\gamma \triangleq \prod_{e \in \gamma} W_e$ and $F_\gamma \triangleq \prod_{e\in\gamma} F_e$ for notational convenience. Finally, let $G_F \triangleq (\mathcal{E}_F, \mathcal{V})$ be the random subgraph of base graph G that has edge set $\mathcal{E}_F \triangleq \{e \in \mathcal{E} | F_e = 1\}$. We denote $\mu_g \triangleq \mathbb{P}[G_F = g]$, and $\eta_\gamma \triangleq \sum_{\{g\in\mathcal{G}|\gamma\in\Gamma(g)\}} \mu_g$. We first provide an explicit characterization of dropout's risk function in (6) in terms of paths in the graph that describes the structure of the NN. This is possible since we assume linear activation functions. The following lemma now holds, and is proved in Appendix F.

Lemma 10 *Assume that the base graph G is a fixed, directed graph without cycles in which all paths have length L and there are d_L output nodes (N6'), that $\sigma(t) = t$ (N7), and that F is independent of (X, Y) (N8). Then*

$$\mathcal{D}(W) = \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{s=1}^{d_L} \big(Y_s - \sum_{\gamma\in\Gamma^s(g)} P_\gamma X_{\gamma_0}\big)^2\Big]. \tag{31}$$

Moreover $\mathcal{D}(W) = \mathcal{J}(W) + R(W)$, where

$$\mathcal{J}(W) = \sum_{\gamma\in\Gamma(G)} \eta_\gamma \mathbb{E}[(Y_{\gamma_L} - P_\gamma X_{\gamma_0})^2], \tag{32}$$

$$R(W) = -\sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{s=1}^{d_L} \sum_{\gamma\in\Gamma^s(g)} \Big(\Big(1 - \frac{1}{|\Gamma^s(g)|}\Big) Y_s^2 - P_\gamma X_{\gamma_0} \sum_{\delta\in\Gamma^s(g)\setminus\{\gamma\}} P_\delta X_{\delta_0}\Big)\Big]. \tag{33}$$

Here, the constants η_γ, μ_γ depend explicitly on F's distribution and the NN's architecture.

Note that Lemma 10 essentially changes variables to rewrite the dropout risk function as a sum over paths instead of a sum over graphs. This representation allows us to clearly identify the regularization term $R(W)$. For example in the case of *Dropconnect* (Wan et al., 2013), where the filter variables $\{F_e\}_{e\in\mathcal{E}}$ are independent random variables with distribution Bernoulli(p), Lemma 10 holds with $\mu_g = p^{|\mathcal{E}(g)|}(1-p)^{|\mathcal{E}(G)|-|\mathcal{E}(g)|}$. Also note that if for all subgraphs $g \in \mathcal{G}$ and vertices $i \in [d]$ the number of paths that end at i satisfies $|\Gamma^i(g)| = 1$, such as when G is an arborescence, then for all subgraphs $g \in \mathcal{G}$ and paths $\gamma \in \Gamma(g)$ there is only one path ending at a leave node γ_L, that is, $\Gamma^{\gamma_L}(g) = \{\gamma\}$.

We now focus on a base graph that is an arborescence of arbitrary depth; see Figure 1c. Hence we now replace (N6') in Lemma 10 that assumes a generic graph by assumption (N6), where G is specifically an arborescence. The following specification of Corollary 11 is also proven in Appendix F.

Corollary 11 *Assume that the base graph G is an arborescence of depth L (N6), and (N7)–(N8) from Lemma 10. Then $\mathcal{D}(W) = \mathcal{I}(W) + \mathcal{D}(W^{\mathrm{opt}})$, where*

$$\mathcal{I}(W) \triangleq \sum_{\gamma \in \Gamma(G)} \nu_\gamma (z_\gamma - P_\gamma)^2,$$

$$\mathcal{D}(W^{\mathrm{opt}}) = \sum_{\gamma \in \Gamma(G)} \eta_\gamma (\mathbb{E}[Y_{\gamma_L}^2] - \mathbb{E}[Y_{\gamma_L} X_{\gamma_0}]^2 / \mathbb{E}[X_{\gamma_0}^2]), \tag{34}$$

and $\nu_\gamma \triangleq \eta_\gamma \mathbb{E}[X_{\gamma_0}^2]$, $z_\gamma \triangleq \mathbb{E}[Y_{\gamma_L} X_{\gamma_0}]/\mathbb{E}[X_{\gamma_0}^2]$ for $\gamma \in \Gamma(G)$. Consequently, $R(W) = 0$ for an arborescence.

The convergence result we are about to show uses the fact that for the system of ODEs $\mathrm{d}W/\,\mathrm{d}t = -\nabla_W \mathcal{D}(W)$ there are conserved quantities. Within the proof, these conserved quantities have the crucial role of guaranteeing compactness for the iterates. Specifically, let $\mathcal{L}(g; f)$ denote the leaves of the subtree of $g \in \mathcal{G}$ rooted at a vertex $f \in \mathcal{E}(g)$, and define the set of leaves of G as $\mathcal{L}(G) \triangleq \cup_{f \in \mathcal{E}} \mathcal{L}(G; f)$. We remark that in the previous notation $d_L = |\mathcal{L}(G)|$. For $W \in \mathcal{W}$ and each leaf $f \in \mathcal{E} \backslash \mathcal{L}(G)$, define the quantity

$$C_f = C_f(W) \triangleq W_f^2 - \sum_{l \in \mathcal{L}(G;f)} W_l^2. \tag{35}$$

Define $C_{\min} \triangleq \min_{e \in \mathcal{E} \backslash \mathcal{L}(G)} C_e$ and $C_e^{\{t\}} = C_e(W^{\{t\}})$ for $t \in \mathbb{N}_+$ also, both of which we require later. Lemma 12 now proves that the function C_f in (35) is a conserved quantity; the proof is in Appendix G.

Lemma 12 *Assume (N2) from Proposition 6, (N6) from Corollary 11 , (N7), (N8) from Lemma 10. Then under the negative gradient flow $\mathrm{d}W/\,\mathrm{d}t = -\nabla \mathcal{D}(W)$,*

$$\frac{\mathrm{d}C_f}{\mathrm{d}t} = 0 \tag{36}$$

for all $f \in \mathcal{E} \backslash \mathcal{L}(G)$.

We are almost in position to state our second result, but need to introduce still some notation. We define the following constants

$$\|\nu\|_1 \triangleq \sum_{\gamma \in \Gamma(G)} \nu_\gamma, \quad \nu_{\min} \triangleq \min_{\gamma \in \Gamma(G)} \nu_\gamma, \quad \nu_{\max} \triangleq \max_{\gamma \in \Gamma(G)} \nu_\gamma, \tag{37}$$

for notational convenience. Also, for $0 < \delta < M$, we define

$$\mathcal{S} \triangleq \{W \in \mathcal{W} \;:\; M > |W_f| > \delta > 0 \;\forall f \in \mathcal{E}(G) \backslash \mathcal{L}(G); M > |W_f| \;\forall f \in \mathcal{L}(G)\}, \tag{38}$$

a bounded set of parameters where if the weight is associated with a leaf, they are furthermore bounded away from zero. Let finally

$$B(\epsilon, I) \triangleq \Big\{ W \in \mathcal{W} \;:\; \mathcal{I}(W) \leq \epsilon, W_f^2 - \sum_{l \in \mathcal{L}(G;f)} W_l^2 \in I_f \text{ for } f \in \mathcal{E} \backslash \mathcal{L}(G) \Big\} \tag{39}$$

denote the set of all weight parameters that are ε-close to a critical point and for which the conserved quantities in (35) deviate by no more than $O(C_f^{\{0\}})$ from their initial value $C_f^{\{0\}}$. These deviations are made explicit by the intervals

$$I_f \triangleq [C_f^{\{0\}}/2, 3C_f^{\{0\}}/2] \text{ for } f \in \mathcal{E}\backslash\mathcal{L}(G), \text{ and the set } I \triangleq \times_{f\in\mathcal{E}\backslash\mathcal{L}(G)} I_f \subseteq \mathbb{R}^{|\mathcal{E}|-|\mathcal{L}(G)|}. \tag{40}$$

Our proof shows that the iterates $\{W^{\{t\}}\}_{t\geq 0}$ stay in the intersection $\mathcal{S} \cap B(\varepsilon, I)$, and this implies that the weights (including those associated with the leaves) remain bounded. The following now holds, and its proof can be found in Appendix H.

Proposition 13 *Assume (N2) from Proposition 6, (N6) from Corollary 11, (N7)–(N8) from Lemma 10, that $W^{\{0\}} \in \mathcal{S} \cap B(\epsilon, I)$ and $M^L \geq |z_\gamma|$ for all $\gamma \in \Gamma(G)$ (N9), that $\frac{1}{2}C_{\min}(W^{\{0\}}) > \delta^2$ (N10). If*

$$\alpha \leq \min\Big(\nu_{\min}\frac{e^{1/2}(C_{\min}^{\{0\}})^L}{16\left\|\nu\right\|_1 LM^{2(L-1)}\mathcal{I}(W^{\{0\}})}, \frac{1}{12\nu_{\max}\left|\mathcal{E}\right|\left|\Gamma(G)\right|M^{2(L-1)}}, \frac{1}{2\nu_{\min}(C_{\min}^{\{0\}})^{L-1}}\Big), \tag{41}$$

then the iterates of (27) *satisfy*

$$\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\text{opt}}) \leq \big(\mathcal{D}(W^{\{0\}}) - \mathcal{D}(W^{\text{opt}})\big)\exp(-\tfrac{\alpha\tau}{2}t). \tag{42}$$

where $\tau = 4\nu_{\min}\exp(-1/2)(C_{\min}^{\{0\}})^{L-1}$.

Proposition 13 identifies explicitly how the convergence rate of GD on a dropout's risk function depends on the dropout algorithm and the structure of the arborescence: parameters such as $p, |\mathcal{L}(G)|, L$ are implicitly present in the constants $\nu_{\min}$ and $\|\nu\|_1$ in α, τ.

Note that Assumptions (N9)–(N10) are relatively benign. These assumptions are for example satisfied when initializing $M > W_e^{\{0\}} > \sqrt{2}\delta$ for $e \in \mathcal{E}\backslash\mathcal{L}(G)$ and setting $|W_l| \leq \delta/\sqrt{|\mathcal{L}(G)|}$ for all $l \in \mathcal{L}(G)$ and $\epsilon = \mathcal{I}(W^{\{0\}})$, which we assume in Proposition 9. In other words, this initialization sets the weights that are associated with leaves small compared to all other weights.

5. Effect of Dropout on the Convergence Rate in Wider Networks

In Proposition 13, we have proven that the convergence rate depends on p^L for NNs shaped like arborescences. Let G_{tree} be a tree and $e \in \mathcal{E}(G_{\text{tree}})$ be an edge. Denote by $\Gamma^{[t]}(e)$ the set of paths passing through e that are not filtered by dropout at time t. We observe that at any given time t of dropout SGD,

$$\mathbb{P}[w_e^{[t]} \text{ is updated}] = \mathbb{P}[\Gamma^{[t]}(e) \neq \emptyset] = p^L. \tag{43}$$

If we denote by $t_{\text{update}}(G_{\text{tree}}) = 1/p^L$ the average update time for a weight in G_{tree}, then we need $1/p^L$ more time on average for a given edge to be updated than when we do not use dropout. For wider networks G, however, edges can be updated simultaneously and repeatedly via different available paths. By the previous intuition we might still expect that, if the updates are sufficiently independent, the convergence rate depends approximately on $1/t_{\text{update}}$. In order to verify this intuition we will determine t_{update} for NNs that are much wider than deep, and later simulate their convergence rates also in realistic settings.

Suppose now that G is a graph of a fully-connected NN with L dropout layers each of which has width D. For each of the vertices $u \in G$ in a dropout layer, there is an associated dropout filter variable $F_u \sim_{\text{i.i.d.}} \text{Ber}(p)$ where $p > 0$ is fixed. That is, we use *dropout.* Note that any other additional input or output layer without filters only changes the number of paths by a multiplicative factor. Hence, we will restrict to the case that all nodes in the layers have filter variables. In this case, we may consider a path $\gamma = (u_1, \ldots, u_L)$ as a set of L vertices—one for each dropout layer—instead of edges. For two paths γ and δ, we consider their intersection $\gamma \cap \delta$ as the subset of vertices belonging to both paths. Hence, $|\gamma \cap \delta| = l$ implies that the intersection has l vertices, not necessarily forming a path.

We remark that we can restrict to the case $L > 2$. In the case of one dropout layer $L = 1$, an edge $e = (u, v)$ conected to a dropout node u is updated if and only if the filter $F_u = 1$, where $u \in G$ is the adjacent vertex to e with a dropout filter, so that in this case $\mathbb{P}[w_e^{[t]}$ is updated$] = 1 - p$. For $L = 2$, an edge $e = (u, v)$ is updated if and only if $F_u = F_v = 1$, so that $\mathbb{P}[w_e^{[t]}$ is updated$] = 1 - p^2$. Recall that we denote by $\Gamma(e)$ the set of paths γ of G passing through e. For a path $\gamma \in \Gamma(e)$, in the following, we let $F_\gamma = \prod_{u \in \gamma} F_u$ be the indicator of a path being filtered. Thus, F_γ is 1 is γ is not filtered and 0 otherwise. We will use Greek letters for paths and Latin letters for vertices when referring to filters F_γ and F_u respectively.

Lemma 14 *Let G be a graph of a fully-connected NN with $L > 2$ dropout layers, each with the same width D and with dropout filters F_u for $u \in G$. For an edge $e \in \mathcal{E}(G)$, let $F_{\Gamma(e)} = \sum_{\gamma \in \Gamma(e)} F_\gamma$ denote the random variable that counts the number of nonfiltered traversing paths through e. If L, p are fixed, then as $D \to \infty$,*

$$\mathbb{P}[F_{\Gamma(e)} = 0] = 1 - p^2 + O\Big(\frac{pL}{D}\Big). \tag{44}$$

Proof We will use the Paley–Zygmund inequality. For a nonnegative random variable Z with finite second moment, for any $\theta \in (0, 1)$,

$$\mathbb{P}[Z > \theta \mathbb{E}[Z]] \geq (1 - \theta)^2 \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]}. \tag{45}$$

We will use (45) with the random variable $F_{\Gamma(e)}$. The idea is that if D is much larger than L, the average number of paths passing through e is also large. We are using dropout, so the filter variable corresponding to an edge $e = (u, v)$ will depend on the vertex u only, that is, $F_e = F_u$. For counting paths we also need to take into account that the filter F_v will appear in all paths passing through e. Since only the two vertices u and v of e are fixed we can compute

$$\mathbb{E}[F_{\Gamma(e)}] = \sum_{\gamma \in \Gamma(e)} \mathbb{E}[F_\gamma] = p^L |\Gamma(e)| = p^L D^{L-2}. \tag{46}$$

We define the set of broken paths in $\Gamma(e)$ as

$$\Gamma_b(e) = \{\gamma = (u_{i_1}, \dots, u_{i_k}) \in G^k : \exists \eta, \delta \in \Gamma(e), \gamma = \eta \cap \delta\}, \tag{47}$$

that is, $\gamma \in \Gamma_b(e)$ if and only if there exist $\eta, \delta \in \Gamma(e)$ such that $\gamma = \eta \cap \delta$. In particular, $\Gamma_b(e)$ contains paths and unions of vertices of paths that pass through e. Then we have:

$$\begin{aligned}
\mathbb{E}[F_{\Gamma,e}^2] &= \sum_{\gamma\in\Gamma(e)} \sum_{\delta\in\Gamma(e)} \mathbb{E}[F_\gamma F_\delta] \overset{\text{(i)}}{=} \sum_{\gamma\in\Gamma(e)} \sum_{l=2}^{L} \sum_{\substack{\delta\in\Gamma(e)\\ |\gamma\cap\delta|=l}} \mathbb{P}[F_\gamma = 1, F_\delta = 1] && (48)\\
&\overset{\text{(ii)}}{=} \sum_{\gamma\in\Gamma(e)} \sum_{l=2}^{L} \sum_{\substack{\delta\in\Gamma(e)\\ |\gamma\cap\delta|=l}} p^l p^{2L-2l} \overset{\text{(iii)}}{=} \sum_{l=2}^{L} \sum_{\substack{\eta\in\Gamma_b(e)\\ |\eta|=l}} \sum_{\substack{\gamma,\delta\in\Gamma(e)\\ \eta\subseteq\delta,\gamma\\ \gamma\cap\delta=\eta}} p^l p^{2L-2l} && (49)\\
&\overset{\text{(iv)}}{=} \sum_{l=2}^{L} \sum_{\substack{\eta\in\Gamma_b(e)\\ |\eta|=l}} (D(D-1))^{L-l} p^l p^{2L-2l} && (50)\\
&\overset{\text{(v)}}{=} \sum_{l=2}^{L} \binom{L-2}{l-2} D^{l-2} (D(D-1))^{L-l} p^l p^{2L-2l} && (51)\\
&= p^{2L-2} D^{2L-4} + O(Lp^{2L-3} D^{2L-5}), && (52)
\end{aligned}$$

where (i) we have first used that F_γ are indicators for occurring $\gamma \in \Gamma(e)$ and that at least $l \geq 2$ since vertices u and v are shared among all paths in $\Gamma(e)$; secondly, that we have separated the sum over paths into a path γ and all other paths δ that coincide in l vertices. In (ii) we have computed the probability by noting that for γ and δ such that $|\gamma \cap \delta| = l \geq 2$, $\mathbb{E}[F_\gamma F_\delta] = p^l p^{2L-2l}$, where the term p^l accounts for the l shared filters corresponding to l shared vertices and p^{2L-2l} for the remaining products of filters. Note that we have used the independence assumption for filters here. (iii) We have used here that $\eta = \delta \cap \gamma \in \Gamma_b(e)$, so that we can separate the previous sum into first, fixing the l vertices where two paths intersect—including e—with $\eta \in \Gamma_b(e)$ such that $|\eta| = l$, and then looking for all possible $\delta, \gamma \in \Gamma(e)$ such that $\gamma \cap \delta = \eta$. For (iv) we fix l vertices where γ and δ coincide, then there are still $(D(D-1))^{L-l}$ possible ordered vertex pairs to choose from all the other vertices where γ and δ do not coincide. (v) For the remaining sum, for each l fixed locations—including the vertices of e, which are fixed—we can still choose D^{l-2} remaining possible vertices. Additionally, there are for each l, $\binom{L-2}{l-2}$ distinct $l-2$ locations for these vertices. Hence, plugging (52) and (46) into (45) yields

$$\begin{aligned}
\mathbb{P}[F_{\Gamma(e)} > \theta p^L D^{L-2}] &\geq (1-\theta)^2 \frac{p^{2L} D^{2L-4}}{p^{2L-2} D^{2L-4} + O(Lp^{2L-3} D^{2L-5}))} && (53)\\
&= (1-\theta)^2 \frac{p^2}{1 + O(L/(Dp))} && (54)\\
&= (1-\theta)^2 (p^2 + O(pL/D)). && (55)
\end{aligned}$$

In particular, setting $\theta^{-1} = 2p^L D^{L-2}$ and computing the higher order noting that $L > 2$, we obtain that

$$\mathbb{P}[F_{\Gamma(e)} > 1/2] \geq p^2 + O(pL/D), \tag{56}$$

or alternatively noting that $\{F_{\Gamma(e)} \leq 1/2\} = \{F_{\Gamma(e)} = 0\}$, since $F_{\Gamma(e)} \in \mathbb{N}$ we obtain

$$\mathbb{P}[F_{\Gamma(e)} = 0] \leq 1 - p^2 + O(pL/D). \tag{57}$$

Finally note that $1 - p^2 \leq \mathbb{P}[F_{\Gamma(e)} = 0]$ since the edge e can be present in a path only if the filters at both vertices of e have value 1, which occurs with probability p^2, so that $\mathbb{P}[F_{\Gamma(e)} > 0] < p^2$. ■

Note that in the proof of Lemma 14 we can recover the scaling p^L that we have seen in Proposition 13 by setting $D = 1$ in (52) and in (50).

From Lemma 14 we expect that for a wide network with L layers where $D \gg L$ and an edge $e \in \mathcal{E}(G)$, we have that

$$\mathbb{P}[w_e^{[t]} \text{ is updated}] = p^2 + O(pL/D). \tag{58}$$

If the convergence rate is related to the update rule, then we would expect that for a wide network the rate would be independent of L which is different from the path network considered in Proposition 13. In the next section we will verify this intuition on real data sets. Note, however, that we do not expect to see the dependence on p as shown in (58): this heuristic argument provides only the rate at which a weight is updated, and stochastic averaging is not solely driving the convergence rate. In particular, from an example for wide shallow linear networks in Senen-Cerda and Sanders (2022), close to a critical point of a dropout ODE, the dependence scales with a factor $p(1-p)$ instead of p. This is due to the fact that for larger p, there are regions of the landscape close to minima that become flat, as also hinted by Proposition 7. Indeed, when $p \uparrow 1$ the term $(1-p)J \downarrow 0$ in the convergence rate of Proposition 7 lowers the complexity of finding an ϵ-stationary point. Hence, there are landscape regimes and initialization issues that also account for the convergence rate in NNs.

Finally the results from Lemma 14 and Proposition 13 suggests some practical rules for choosing p to avoid a slow empirical risk minimization. Namely, when training networks with width D and L dropout layers such that $D \gg L$, the dependence of the convergence on p may only become slow when choosing a very small p. For networks with L larger than D, however, p needs to be choosen carefully to avoid a compounding effect of dropout that may make the dependence of the convergence rate highly dependent on p as happens with arborescences—the extreme case with $D \simeq 1$.

5.1 Numerical Experiments

In this section we conduct the dropout stochastic gradient descent algorithm numerically,[4] for different data sets and network architectures. We measure the convergence rate for

[4] The source code of our implementation is available at `https://gitlab.tue.nl/20194488/almost-sure-convegence-of-dropout-algorithms-for-neural-networks`.

different widths D, depths L, and dropout probabilities $1-p$. We then compare these measurements to the bounds on the convergence rates obtained in Section 4. We use *Tensorflow*[5] for the implementation. We remark that differently to the theoretical results of Section 3, we will not project the iterates of the gradient descent onto a compact set as with the appropiate initialization we do not observe any diverging trajectory of the iterates.

5.1.1 Setup

Data sets. We will consider three commonly used data sets of images: the MNIST[6] (LeCun et al., 2010), CIFAR-100-fine[7], and CIFAR-100-coarse data sets (Krizhevsky, 2009).
NN Architecture. We use as a base architecture a LeNet with 11 layers where the two dense layers have been substituted with L fully-connected ReLU layers of width D. Each of these layers have dropout with dropout probability $1-p$. While larger networks are commonly used in practice, a LeNet architecture is sufficient to test the effect of dropout on the convergence rate as we verify with the simulations.
Loss. We use the cross-entropy loss, which is commonly used for classification. For two distributions p and q with support on $[n]$ labels, the cross-entropy loss is defined as

$$l(p,q) = -\sum_{i=1}^{n} q_i \log(p_i). \tag{59}$$

Stopping criteria. In all experiments, we stop after 40 epochs.
Initialization. In order to see the convergence rate close to a minimum. We use first a *Gaussian initialization*, that is, we set every weight on the dense layers to $W_{ijk} \sim \mathrm{Normal}(0, 1/\sqrt{D})$ in an independent manner, where D is the width of the layer. While this initialization is standard, we note that we cannot expect to compare convergence rates for different numbers of layers $L \in \{1,2,3\}$ and for different dropout probabilities $1-p$, since the loss functions are also different. In the course of our experiments, we found that there are also many saddle points where SGD remains stuck, which complicated the estimation of the convergence rate. In order to start approximately at the same neighborhood where the iterates stay and continuously track minima across different choices of p, for each $L \in \{1,2,3\}$ we have used a two-step approach in order to avoid areas of the landscape with saddle points. We first run ADAM[8] for 2 epochs with $p = 0.1$ and store the weights. Secondly, for each $p \in P$ we then perform dropout SGD with initialization given by the stored weights. In this manner, we expect that we are approximately “tracking” the same local region across the optimization landscape when we change p. Optimization with ADAM is less prone to remain in flat areas of the landscape since it uses a dynamic step size. Hence, if after the dynamic step the iterates remain in a part of the landscape with no saddle points that smoothly changes with p, we also expect in this case to obtain comparable convergence rates for SGD for each fixed L.
Step size and batch size. In each experiment, the step size is given by $\eta = 10^{-5}$ and the batch size is $b = 1024$. Both have been chosen by hand for convenience and no additional

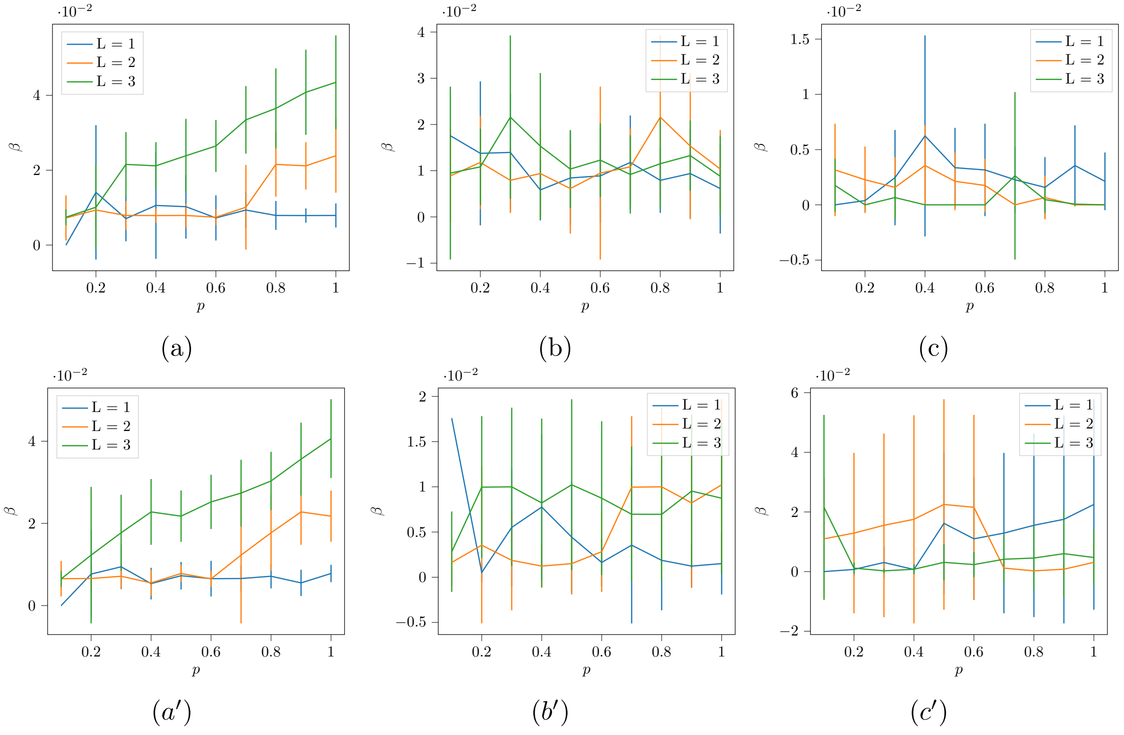
[5] `https://www.tensorflow.org/`
[6] Modified National Institute of Standards and Technology (MNIST)
[7] Canadian Institute For Advanced Research (CIFAR)
[8] Adaptative Moment Estimation (See Kingma and Ba (2014)).

fine-tuning has been conducted for the experiment. A small batch-size compared to the data set size has been selected to allow for a stochastic trajectory to occur.

Fitting procedure. We fix a set of probabilities $P \subset [0,1]$ and depths $L = \{1,2,3\}$ and for each pair $(p,l) \in P \times L$ we run the algorithm above. From the value of the loss from all T iterations of SGD $\mathcal{L} = (l_t)_{t=0}^T$ in one run, we compute a moving average $a(\mathcal{L})_{t=0}^T$, where we average the loss across a window with size given by the number of batches n_b required to complete one epoch. In this manner we obtain an average convergence rate and diminish the stochasticity from the data set. We then fit the averaged loss of the iterates $a(\mathcal{L})_{t=0}^T$ for each p and l to the function

$$f(\alpha_{p,l}, \beta_{p,l}, \gamma_{p,l}) = \alpha_{p,l} \exp(-\beta_{p,l} t) + \gamma_{p,l}. \tag{60}$$

We run the experiment $R = 10$ times for each (p,l) and obtain an average convergence exponent $(\tilde{\beta}_{p,l})_{(p,l)\in P\times L}$.



(a) (b) (c)

(a') (b') (c')

Figure 3: The fit $\tilde{\beta}_{p,l}$ for $p \in \{i \times 10^{-1} : i \in [10]\}$ and $l \in \{1,2,3\}$ for LeNet with different widths D and different data sets. Here *(a)* MNIST with $D = 50$; *(a′)* MNIST with $D = 100$; *(b)* CIFAR-100-fine labels with $D = 50$; *(b′)* CIFAR-100-fine labels with $D = 100$; *(c)* CIFAR-100-coarse labels with $D = 50$; (c') CIFAR-100-coarse labels with $D = 100$. While for the MNIST data set there seems to be an increasing dependence of dropout on the convergence rate with the depth L, for CIFAR no such dependence is observed. We remark, however, that in the CIFAR data sets encountering saddle points was more common. For those areas the loss profile is flat and so we expect the fits to be biased towards the origin in some cases.

5.1.2 Results

In Figure 3 we can see the plots of $\tilde{\beta}_{p,l}$. As suspected from the heuristic argument, we do not see an increasingly large dependence on p for $L = 1, 2$ or 3 when $D \in \{50, 100\}$. For the MNIST data set some dependence on the depth is appreciated, but this may be due to other factors that affect the convergence rate, like initialization issues. For the CIFAR data sets, convergence is greatly affected by saddlepoints despite the use of dropout. This is, however, common when using SGD with small constant stepsizes. In particular, in practical scenarios other schemes that adjust the stepsize, like e.g. ADAM, may be more appropriate when dealing with deep networks with dropout in different layers. From the experiments it is concluded that despite the stochasticity provided by dropout, the convergence rate is not affected much by a varying dropout probability $1 - p$ in wide networks with just few dropout layers.

6. Conclusion

Firstly, this paper contains a probability theoretical proof that a large class of dropout algorithms for neural networks converge almost surely to a unique stationary set of a projected system of ODEs; see Proposition 6. The result guarantees formally that these dropout algorithms are well-behaved for a wide range of NNs and activation functions, and will at least asymptotically not suffer from issues because of the connection to bond percolation.

Secondly, this paper contains bounds for the sample complexity of SGD with dropout to converge to an ϵ-stationary point of a generic nonconvex function. These can be found in Propositions 7 and 8. An upper bound to the rate of convergence of GD on the limiting ODE of dropout algorithms is also established for arborescences of arbitrary depth with linear activation functions; see Proposition 13. This result is a necessary step towards analyzing the convergence rate of the actual stochastic implementations of dropout algorithms.

For example, note that Proposition 9, which is a consequence of Proposition 13, implies that *Dropout* and *Dropconnect* can impair the convergence rate by an exponential factor in the number of layers of thin but deep networks.

We have theoretically and experimentally verified this impairment in experiments with a path network; see Sections 5.1 and 4. This fact is contrasted though with our experimental nonobservation of a strong dependence on the dropout probability p in wide networks with just a few dropout layers. These two observations together imply that there is a change of regime in the convergence rate from networks that are wide with a few dropout layers to thin networks with many dropout layers.

6.1 Future Research

In the first half of this paper, we relied on the ODE method. Observe specifically that we used it to study the limiting behavior of Dropout and/or Dropconnect when the number of SGD iterations becomes large and the topology of the NN is kept fixed. Our conclusions imply that ultimately, after an infinite number of iterations on a fixed topology, whether convergence happens is not affected by p bounded away from 0 and 1. However, consider now the following thought experiment for any p bounded away from 0 and 1: keep the number of iteration steps fixed and grow instead the NN infinitely deep. No information can then propagate from the input layer to the output layer with probability one due to the percolation phenomenon. Consequently, there must be an intermediate scaling regime—one in which both time and space are scaled simultaneously—to which the ODE method can potentially be applied, and in which percolation affects whether dropout converges. Precisely this warrants further study.

The generality of the sample complexity result in Section 3, as well as the heuristic arguments of Section 5 suggest that dropout increases the complexity of learning from empirical data when using dropout SGD. For a fixed training budget, an interesting follow-up question is what is the benefit of increasing the complexity in training that comes from dropout regularizing and improving the generalization capabilities of the trained model. In this question is then implicit what is the best choice for p that makes the training with dropout optimal given a training budget.

Acknowledgments

We thank the anonymous referees for their feedback. Their suggestions have led to an improved paper.

Appendix

Appendix A. Backpropagation Algorithm

We define the backpropagation algorithm used in Section 2 to compute the estimate of the gradient.

Definition 15 *Assume $\sigma \in C^1(\mathbb{R})$. Given weights $W \in \mathcal{W}$ and input–output pair $(x, y) \in \mathbb{R}^{d_0} \times \mathbb{R}^{d_L}$, the tensor $\mathrm{B}_W(x, y) \in \mathbb{R}^{d_L \times d_{L-1}} \times \cdots \times \mathbb{R}^{d_1 \times d_0}$ is calculated iteratively by:*

1. *Computing $A_1, \dots, A_L$ using Definition 4.*
2. *Calculating for $i = L-1, \dots, 1$,*

$$\begin{aligned} R_L &= A_L = (y - W_L A_{L-1}) \in \mathbb{R}^{d_L}, \\ R_i &= (W_{i+1}^{\mathrm{T}} R_{i+1}) \odot (\sigma'(W_i A_{i-1})) \in \mathbb{R}^{d_i}. \end{aligned} \tag{61}$$

3. *Setting for $i \in [L]$, $\big(\mathrm{B}_W(x, y)\big)_i = -2 R_i A_{i-1}^{\mathrm{T}}$.*

Definition 15 is essentially a computationally efficient manner of calculating the gradient $\nabla l(\Psi_W(x), y)$ in (1), leveraging the NN's layered structure together with the chain rule of differentation to come to a recursive computation of the partial derivatives.

Appendix B. ODE Method

Regarding our second result in Proposition 13, observe that GD on a limiting ODE is not exactly a dropout algorithm. Analyzing GD's convergence rate however is an important stepping stone towards analyzing the convergence rate of dropout algorithms. To see the mathematical relation, consider that any dropout algorithm updates the weights

$$W^{[n+1]} = W^{[n]} + \alpha^{\{n\}} \Delta^{[n+1]} \tag{62}$$

randomly for $n = 0, 1, 2, \cdots$. Here, the $\alpha^{\{n\}}$ denote the step sizes of the algorithm, and the $\Delta^{[n+1]}$ represent the random directions that result from the act of dropping weights. As we will show in this paper under assumptions of independence, these random directions satisfy

$$\mathbb{E}[\Delta^{[n+1]} \mid W^{[0]}, \dots, W^{[n]}] = -\nabla \mathcal{D}(W^{[n]}) \tag{63}$$

for some continuous, differentiable function $\mathcal{D}(W)$. Observe that the algorithm in (62) satisfies $W^{[n+1]} = W^{[n]} + \alpha^{\{n\}}(-\nabla \mathcal{D}(W^{[n]}) + M^{[n+1]})$ where $M^{[n+1]} = \mathbb{E}[\Delta^{[n+1]} \mid W^{[0]}, \dots, W^{[n]}] - \Delta^{[n+1]}$ describes a *martingale difference* sequence. This martingale difference sequence's expectation with respect to the past $W^{[0]}, \dots, W^{[n]}$ is zero.

For diminishing step sizes $\alpha^{\{n\}}$, we can consequently view dropout algorithms as in (62) as being noisy discretizations of the ordinary differential equation

$$\frac{\mathrm{d}W}{\mathrm{d}t} = -\nabla \mathcal{D}(W(t)). \tag{64}$$

In fact, we employ the so-called *ordinary differential equation method* (Kushner and Yin, 2003; Borkar, 2009), which formally establishes that the random iterates in (62) follow the trajectories of the gradient flow in (64). Hence, after sufficiently many iterations n and for a sufficiently small step size α, the convergence rate of the deterministic GD algorithm

$$W^{\{n+1\}} = W^{\{n\}} - \alpha \nabla \mathcal{D}(W^{\{n\}}) \tag{65}$$

gives insight into the convergence rate of the stochastic dropout algorithm in (62).

Appendix C. Projection Operator

We define here the projection operator π used in Section 3. Say that $\mathcal{H}$ is defined by l smooth constraints $q_i : \mathcal{W} \to \mathbb{R}$, $i = 1, \dots, l$ satisfying $q_1(W) \leq 0, \dots, q_l(W) \leq 0$, i.e., $\mathcal{H} = \{W \in \mathcal{W} : q_i(W) \leq 0 \ \forall i \in [l]\}$. Denote by $\nabla \mathcal{D}|_{\mathcal{H}}(W)$ the gradient of $\mathcal{D}(W)$ restricted to $\mathcal{H}$ and let $\mathrm{T_W}\mathcal{W}$ be the tangent space of $\mathcal{W}$ at W. Suppose that $\nabla q_i(W) \neq 0$ whenever $q_i(W) = 0$, and that these are linearly independent. At any point $W \in \partial \mathcal{H}$, we define the outer normal cone

$$C(W) \triangleq \{v \in \mathrm{T_W}\mathcal{W} \ : \ \nabla q_i(W) v^T \geq 0 \text{ for } i \in [l] \text{ s.t. } q_i(W) = 0\}. \tag{66}$$

We also assume that $C(W)$ is upper semicontinuous, i.e., if $\tilde{W} \in B_{\mathcal{H}}(W, \delta)$, where $B_{\mathcal{H}}(W, \delta)$ is the ball of radius $\delta > 0$ centered at W and intersected with $\mathcal{H}$, then $C(W) = \cap_{\delta > 0} \big(\cup_{\tilde{W} \in B_{\mathcal{H}}(W,\delta)} C(\tilde{W})\big)$. Let $\pi(W) \triangleq -t\mathbb{1}[W \in \partial \mathcal{H}]$ with $t \in C(W)$ minimal to resolve the violated constraints of $\mathcal{D}|_{\mathcal{H}}(W)$ at $W \in \partial \mathcal{H}$ so that $\mathcal{D}|_{\mathcal{H}}(W) + \pi(W)$ points inside $\mathcal{H}$. In particular, we have

$$\pi(W) = -\sum_{i=1}^{l} \lambda_i(W) \nabla q_i(W) \in -C(W) \tag{67}$$

where $\{\lambda_i(W) \geq 0\}_{i=1}^{l}$ are functions such that $\lambda_i(W) = 0$ if $q_i(W) < 0$.

Appendix D. Proof of Proposition 6

The proof of Proposition 6 relies on the framework of stochastic approximation in Kushner and Yin (2003). Specifically, Proposition 6 follows from Theorem 2.1 on p. 127 if we can show that its conditions (A2.1)–(A2.6) on p. 126 are satisfied. In the notation of Sections 2, 3, these conditions read:

(A2.1) $\sup_t \mathbb{E}[\|\Delta^{[t+1]}\|_{\mathrm{F}}] < \infty$;

(A2.2) there is a measurable function $\bar{g}(\cdot)$ of W and there are random variables $\beta^{[t+1]}$ such that

$$\mathbb{E}[\Delta^{[t+1]} \mid \mathcal{F}_t] = \bar{g}(W^{[t]}) + \beta^{[t+1]}, \tag{68}$$

where $\mathcal{F}_t$ denotes the smallest σ-algebra generated by $\cup_{s\leq t}\{W^{[0]}, (F^{[s]}, X^{[s]}, Y^{[s]})\}$;

(A2.3) $\bar{g}(\cdot)$ is continuous;

(A2.4) the step sizes satisfy

$$\sum_{t=1}^{\infty} \alpha^{\{t\}} = \infty, \alpha^{\{n\}} \geq 0, \alpha^{\{n\}} \to 0 \text{ for } n \geq 0 \text{ and } \alpha^{\{n\}} = 0 \text{ for } n < 0; \tag{69}$$

$$\sum_{t=1}^{\infty} (\alpha^{\{t\}})^2 < \infty; \tag{70}$$

(A2.5) $\sum_t \alpha^{\{t\}} \|\beta^{[t]}\|_{\mathrm{F}} < \infty$ w.p. one;

(A2.6) $\bar{g}(\cdot) = -\nabla \mathcal{D}(\cdot)$ for a continuously differentiable real-valued $\mathcal{D}(\cdot)$ and $\mathcal{D}(\cdot)$ is constant on each stationary set S_i.

We next also state for your convenience Theorem 2.1 by Kushner and Yin (2003) in the notation of this paper. Their result does require some notation, as it characterizes the limiting behavior of the iterates of

$$W^{[n+1]} = \mathcal{P}_{\mathcal{H}}\big(W^{[n]} - \alpha\Delta^{[n+1]}\big) \triangleq W^{[n]} - \alpha\Delta^{[n+1]} + Z^{[n+1]}. \tag{71}$$

For any sequence of step sizes $\alpha^{\{n\}}$ satisfying (A2.4), define $t_0 = 0$ and $t_n = \sum_{i=0}^{n-1} \alpha^{\{i\}}$. Define the continuous-time interpolation

$$W_0(t) = \begin{cases} W^{[n]} & \text{for} \quad t_n \leq t < t_{n+1}, \\ W^{[0]} & \text{for} \quad t \leq 0, \end{cases} \tag{72}$$

as well as for $m \in \mathbb{N}_0$, the shifted processes $W_m(t) = W_0(t_m + t)$ for $t \in (-\infty, \infty)$. Let furthermore $o(t) = \inf\{n \in \mathbb{N}_0 : t_n \leq t < t_{n+1}\}$ for $t \in [0, \infty)$, and $o(t) = 0$ for $t \in (-\infty, \infty)$, and define

$$Z_0(t) = \begin{cases} \sum_{i=0}^{o(t)-1} \alpha^{\{i\}} Z_i & \text{for} \quad t \in [0, \infty), \\ 0 & \text{for} \quad t \in (-\infty, \infty), \end{cases} \tag{73}$$

as well as for $m \in \mathbb{N}_0$, the shifted processes $Z_m(t) = \sum_{i=m}^{o(t_m+t)-1}$ for $t \in [0, \infty)$ and $Z_m(t) = -\sum_{i=o(t_m+t)}^{m-1} \alpha^{\{i\}} Z_i$ for $t \in (-\infty, 0)$. The following now holds:

Theorem 16 (A part of Theorem 2.1 by Kushner and Yin (2003)) *Let conditions (A2.1)–(A2.5) hold for algorithm* (71)*, with the projection onto $\mathcal{H}$ being as described in Appendix C. Then there is a set N of probability zero such that for $\omega \notin N$, the set of functions $\{W_m(\omega, \cdot), Z_m(\omega, \cdot), m < \infty\}$ is equicontinuous. Let $(W(\omega, \cdot), Z(\omega, \cdot))$ denote the limit of some convergent subsequence. Then this pair satisfies the projected ODE* (16)*, and $\{W^{[n]}(\omega)\}$ converges to some limit set of the ODE in $\mathcal{H}$. Suppose that (A2.6) holds. Then, for almost all ω, $\{W^{[n]}(\omega)\}$ converges to a unique S_i.*

In order to apply Theorem 16 and arrive at Proposition 6, we verify conditions (A2.1)–(A2.6) through Lemmas 17–19 shown next in Appendix D.1. These lemmas are proven in Appendices D.1.1–D.1.3, respectively.

D.1 Verification of Conditions (A2.1)–(A2.6)

First we assume conditions (N1)–(N3) and we prove that the variance of the random update direction in (4) is finite. This verifies condition (A2.1). The proof can be found in Appendix D.1.1:

Lemma 17 *Assume (N1)–(N3) from Proposition 6. Then* $\sup_{t\in\mathbb{N}} \mathbb{E}[\|\Delta_i^{[t+1]}\|_{\mathrm{F}}^2] < \infty$ *for* $i = 0, 1, \ldots, L$.

We prove next that if $\sigma \in C^r_{PB}(\mathbb{R})$, then the random update direction in (4), conditional on all prior updates, has conditional expectation $\nabla\mathcal{D}(W^{[t]})$. Lemma 18 verifies conditions (A2.2), (A2.3), and (A2.5) (in particular, here $\beta^{[t]} = 0$). The proof can be found in Appendix D.1.2:

Lemma 18 *Assume (N2)–(N4) from Proposition 6. Then* $\mathbb{E}[\Delta^{[t+1]}|\mathcal{F}_t] = \nabla\mathcal{D}(W^{[t]})$*. Furthermore, $\nabla\mathcal{D} : \mathcal{W} \to \mathcal{W}$ is $r - 1$ times continuously differentiable.*

From these conditions the first part of Proposition 6 follows. To prove the second part of Proposition 6, we have to prove that the set of stationary points $S_{\mathcal{H}}$ is well-behaved in the sense that $\mathcal{D}|_{S_i}(W)$ is constant. If an objective function is sufficiently differentiable, this is guaranteed by the Morse–Sard Theorem (Morse, 1939; Sard, 1942). In the present case however we must take into account the possibility of an intersection of the set of stationary points with the boundary $\partial\mathcal{H}$. Assuming (N4) and (N5) provides sufficient conditions. The proof of Lemma 19 can be found in Appendix D.1.3:

Lemma 19 *If (N2)–(N5) hold, then $\mathcal{D}(W)$ is constant on each S_i.*

Since Conditions (A2.1)–(A2.6) of Thm. 2.1 on p. 127 in Kushner and Yin (2003) are now proven satisfied, the proof of Proposition 6 is now completed. ■

D.1.1 Boundedness of $\Delta^{[t+1]}$ in expectation – Proof of Lemma 17

We need to carefully track all sequences of random variables created by a dropout algorithm throughout this proof, which we state here first explicitly.

Definition 20 (Dropout iterates) *During its* $(t+1)$*-st* feedforward step*, the algorithm iteratively calculates*

$$A_0^{[t+1]} = X^{[t+1]}, \quad A_i^{[t+1]} = \sigma((W_i^{[t]} \odot F_i^{[t+1]})A_{i-1}^{[t+1]}) \tag{74}$$

for $i = 1, 2, \ldots, L-1$*, to output*

$$\Psi_{F^{[t+1]} \odot W^{[t]}}(X^{[t+1]}) = (W_L^{[t]} \odot F_L^{[t+1]})A_{L-1}^{[t+1]} = A_L^{[t+1]}. \tag{75}$$

Subsequently for its $(t+1)$*-st* backpropagation step *the algorithm calculates*

$$\begin{aligned} R_L^{[t+1]} &= (Y^{[t+1]} - (W_L^{[t]} \odot F_L^{[t+1]})A_{L-1}^{[t+1]}) \in \mathbb{R}^{d_L}, \\ R_j^{[t+1]} &= ((W_{j+1}^{[t]} \odot F_{j+1}^{[t+1]})^T R_{j+1}^{[t+1]}) \odot (\sigma'((W_j^{[t]} \odot F_j^{[t+1]})A_{j-1}^{[t+1]})) \in \mathbb{R}^{d_i}, \end{aligned} \tag{76}$$

iteratively for $j = L-1, \ldots, 1$*. The algorithm then calculates*

$$\Delta_i^{[t+1]} = -2F_i^{[t+1]} \odot (R_i^{[t+1]}(A_{i-1}^{[t+1]})^{\mathrm{T}}) \tag{77}$$

for $i = 1, \ldots, L$*, and finally updates all weights according to* (13)*.*

The idea of the proof of Lemma 17 is to expand the terms in $\Delta_i^{[t+1]}$ defined in Definition 20 recursively, and identify a polynomial in variables $\{\|Y\|_2^n \|X\|_2^m\}_{m\in\mathbb{N}_0}$ and $n = 0, 1, 2$. We will use several bounds that pertain to the Frobenius norm, written down in Lemma 30 in Appendix J, and we will iterate these in a moment.

First, we will prove two bounds on the activation function applied to an arbitrary matrix A. Recall that $\sigma \in C_{PB}^2(\mathbb{R})$ by assumption (N1). There thus (i) exists some $C_0, k_0 > 0$ such that $|\sigma(z)| \leq C_0(1+z^2)^{k_0}$ for all $z \in \mathbb{R}$, and there exists some $C_1, k_1 > 0$ such that $|\sigma'(z)| \leq C_1(1+z^2)^{k_1}$ for all $z \in \mathbb{R}$. Let $k = \max\{1, k_0, k_1\}$. Then

$$\|\sigma(A)\|_{\mathrm{F}}^2 = \sum_{i,j} |\sigma(A_{ij})|^2 \overset{\text{(i)}}{\leq} C_0 \sum_{i,j} (1 + A_{ij}^2)^k \overset{\text{(Lemma 30)}}{\leq} C_2(1 + \|A\|_{\mathrm{F}})^{2k} \tag{78}$$

for some constant $C_2 > 0$. Similarly there exists some $C_3 > 0$ such that $\|\sigma'(A)\|_F \leq C_3(1 + \|A\|_{\mathrm{F}})^k$. Note furthermore that (ii) for all $l \geq 0$, by submultiplicativity of the Frobenius norm,

$$\begin{aligned} (1 + \|A\sigma(B)\|_{\mathrm{F}})^l &\overset{\text{(ii)}}{\leq} (1 + \|A\|_{\mathrm{F}} \|\sigma(B)\|_{\mathrm{F}})^l \\ &\overset{(78)}{\leq} \left(1 + C_2^{1/2} \|A\|_{\mathrm{F}} (1 + \|B\|_{\mathrm{F}})^k\right)^l \leq C_4 (1 + \|A\|_{\mathrm{F}})^l (1 + \|B\|_{\mathrm{F}})^{kl} \end{aligned} \tag{79}$$

for $C_4 = \max\{1, C_2^{l/2}\} > 0$. Again, a similar bound holds for σ'.

Next, note that we have by (i) submultiplicativity and Lemma 30 that

$$\|\Delta_i^{[t+1]}\|_{\mathrm{F}} = \|F_i^{[t+1]} \odot (R_i^{[t+1]}(A_{i-1}^{[t+1]})^{\mathrm{T}})\|_{\mathrm{F}} \overset{\text{(i)}}{\leq} \|F_i^{[t+1]}\|_{\mathrm{F}} \|R_i^{[t+1]}\|_{\mathrm{F}} \|A_{i-1}^{[t+1]}\|_{\mathrm{F}}. \tag{80}$$

The first term is bounded with probability one: $F_{i,r,l}^{[t]} \in \{0,1\}$ for all i, r, l, t. For the second term, consider the following bound:

$$\begin{aligned} \|R_i^{[t+1]}\|_{\mathrm{F}} &\overset{(76)}{=} \|(W_{i+1}^{[t]} \odot F_{i+1}^{[t+1]})^{\mathrm{T}} R_{i+1}^{[t+1]} \odot \sigma'\big((W_i^{[t]} \odot F_i^{[t+1]}) A_{i-1}^{[t+1]}\big)\|_{\mathrm{F}} \\ &\overset{(\text{Lemma 30})}{\leq} \|W_{i+1}^{[t]} \odot F_{i+1}^{[t+1]}\|_{\mathrm{F}} \|\sigma'\big((W_i^{[t]} \odot F_i^{[t+1]}) A_{i-1}^{[t+1]}\big)\|_{\mathrm{F}} \|R_{i+1}^{[t+1]}\|_{\mathrm{F}} \end{aligned} \tag{81}$$

for $1 \leq i \leq L$, where we have also used the submultiplicative property. For the third term, consider the next bound: (i) recursing (79) with $A = I$ and $B = (W_j^{[t]} \odot F_j^{[t+1]}) A_{j-1}^{[t+1]}$ etc, we obtain that there exists some $C_5 > 0$, say, so that

$$\begin{aligned} \|A_j^{[t+1]}\|_{\mathrm{F}} &\overset{(74)}{=} \|\sigma((W_j^{[t]} \odot F_j^{[t+1]}) A_{j-1}^{[t+1]})\|_{\mathrm{F}} \overset{(78)}{\leq} C_2(1 + \|(W_j^{[t]} \odot F_j^{[t+1]}) A_{j-1}^{[t+1]}\|_{\mathrm{F}})^k \\ &\overset{(\text{Lemma 30})}{\leq} C_2(1 + \|W_j^{[t]} \odot F_j^{[t+1]}\|_{\mathrm{F}})^k (1 + \|A_{j-1}^{[t+1]}\|_{\mathrm{F}})^k \\ &\overset{(\mathrm{i})}{\leq} C_5\big(1 + \|X^{[t+1]}\|_2\big)^{k^j} \prod_{l=1}^{j-1} \big(1 + \|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{j-l}} \end{aligned} \tag{82}$$

for $j = 1, 2, \ldots, L-1$. Similar to the derivation in (82), we obtain instead with σ' that there exists some $C_6 > 0$ such that

$$\|\sigma'((W_j^{[t]} \odot F_j^{[t+1]}) A_{j-1}^{[t+1]})\|_{\mathrm{F}} \leq C_6\big(1 + \|X^{[t+1]}\|_2\big)^{k^j} \prod_{l=1}^{j-1} \big(1 + \|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{j-l}}. \tag{83}$$

Recall that $\|\Delta_i^{[t+1]}\|_{\mathrm{F}} \leq \|F_i^{[t+1]}\|_{\mathrm{F}} \|R_i^{[t+1]}\|_{\mathrm{F}} \|A_{i-1}^{[t+1]}\|_{\mathrm{F}}$. This, together with using (81) repeatedly for $j = i, \ldots, L-1$, and (82), (83), yields the following inequality

$$\begin{aligned} \|\Delta_i^{[t+1]}\|_{\mathrm{F}} &\overset{(81)}{\leq} \|F_i^{[t+1]}\|_{\mathrm{F}} \|R_L^{[t+1]}\|_{\mathrm{F}} \|A_i^{[t+1]}\|_{\mathrm{F}} \prod_{j=i}^{L-1} \|W_{j+1}^{[t]} \odot F_{j+1}^{[t+1]}\|_{\mathrm{F}} \|\sigma'\big((W_j^{[t]} \odot F_j^{[t+1]}) A_{j-1}^{[t+1]}\big)\|_{\mathrm{F}} \\ &\overset{(82)}{\leq} C_5 \|F_i^{[t+1]}\|_{\mathrm{F}} \big(1 + \|X^{[t+1]}\|_2\big)^{k^i} \prod_{l=1}^{i-1} \big(1 + \|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{i-l}} \\ &\quad \times \|R_L^{[t+1]}\|_{\mathrm{F}} \prod_{j=i}^{L-1} \|W_{j+1}^{[t]} \odot F_{j+1}^{[t+1]}\|_{\mathrm{F}} \|\sigma'\big((W_j^{[t]} \odot F_j^{[t+1]}) A_{j-1}^{[t+1]}\big)\|_{\mathrm{F}} \\ &\overset{(83)}{\leq} C_7 \|F_i^{[t+1]}\|_{\mathrm{F}} \big(1 + \|X^{[t+1]}\|_2\big)^{k^i} \prod_{l=1}^{i-1} \big(1 + \|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{i-l}} \\ &\quad \times \|R_L^{[t+1]}\|_{\mathrm{F}} \prod_{j=i}^{L-1} \|W_{j+1}^{[t]} \odot F_{j+1}^{[t+1]}\|_{\mathrm{F}} \big(1 + \|X^{[t+1]}\|_2\big)^{k^j} \prod_{l=1}^{j} \big(1 + \|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{j-l}} \\ &\leq C_7 \|F_i^{[t+1]}\|_{\mathrm{F}} \|R_L^{[t+1]}\|_{\mathrm{F}} \Big(\prod_{j=i}^{L-1} \|W_{i+1}^{[t]} \odot F_{i+1}^{[t+1]}\|_{\mathrm{F}} \Big) \\ &\quad \times \Big(\prod_{j=i}^{L-1} \big(1 + \|X^{[t+1]}\|_2\big)^{2k^j} \prod_{l=1}^{j} \big(1 + \|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{2k^{j-l}} \Big) \end{aligned}$$

$$
\begin{aligned}
&= C_7 \|F_i^{[t+1]}\|_{\mathrm{F}} \|R_L^{[t+1]}\|_{\mathrm{F}} \Big(\prod_{j=i}^{L-1} \|W_{i+1}^{[t]} \odot F_{i+1}^{[t+1]}\|_{\mathrm{F}} \Big) \\
&\quad \times \big(1 + \|X^{[t+1]}\|_2\big)^{\sum_{j=i}^{L-1} 2k^j} \Big(\prod_{j=i}^{L-1} \prod_{l=1}^{j} \big(1 + \|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{2k^{j-l}} \Big). \qquad (84)
\end{aligned}
$$

Lastly, we bound $\|R_L^{[t+1]}\|_{\mathrm{F}}$. By applying (i) subadditivity of the norm $\|A+B\|_F \leq \|A\|_F + \|B\|_F$ and then using the elementary bound $(a+b)^2 \leq 2(a^2+b^2)$ as well as submultiplicativity, we obtain

$$
\begin{aligned}
&\|R_L^{[t+1]}\|_{\mathrm{F}} \overset{(76)}{=} \|Y^{[t+1]} - (W_L^{[t]} \odot F_L^{[t+1]}) A_{L-1}^{[t+1]}\|_{\mathrm{F}} \qquad (85) \\
&\overset{\text{(i)}}{\leq} \|Y^{[t+1]}\|_2^2 + \|W_L^{[t]} \odot F_L^{[t+1]}\|_{\mathrm{F}} \|A_{L-1}^{[t+1]}\|_{\mathrm{F}} \\
&\overset{(82)}{\leq} \|Y^{[t+1]}\|_2 + \|W_L^{[t]} \odot F_L^{[t+1]}\|_{\mathrm{F}} \big(1 + \|X^{[t+1]}\|_2\big)^{k^{L-1}} \prod_{l=1}^{L-1} \big(1 + 2\|W_l^{[t]} \odot F_l^{[t+1]}\|_{\mathrm{F}}\big)^{k^{L-l}}.
\end{aligned}
$$

By combining inequalities (84), (85), and upper bounding the exponent k^{L-1} of the term $1 + \|X^{[t+1]}\|_{\mathrm{F}}$ in (85) by $2\sum_{j=1}^{L-1} k^j$, we conclude that

$$
\begin{aligned}
&\|\Delta_i^{[t+1]}\|_{\mathrm{F}} \\
&\leq C_8 \|Y^{[t+1]}\|_2 \big(1 + \|X^{[t+1]}\|_2\big)^{2\sum_{j=1}^{L-1} k^j} \|F_i^{[t+1]}\|_{\mathrm{F}} P_1\big(\|W_1^{[t]} \odot F_1^{[t+1]}\|_{\mathrm{F}}, \ldots, \|W_L^{[t]} \odot F_L^{[t+1]}\|_{\mathrm{F}}\big) \\
&\quad + C_9 \big(1 + \|X^{[t+1]}\|_2\big)^{2\sum_{j=1}^{L} k^j} \|F_i^{[t+1]}\|_{\mathrm{F}} P_2(\|W_1^{[t]} \odot F_1^{[t+1]}\|_{\mathrm{F}}, \ldots, \|W_L^{[t]} \odot F_L^{[t+1]}\|_{\mathrm{F}}) \qquad (86)
\end{aligned}
$$

for $i = 1, \ldots, L$ and some constants C_8, C_9 and polynomials $P_1(z_1, \ldots, z_L), P_2(z_1, \ldots, z_L)$, say, the latter both in L variables. Because of the projection and by definition of $\mathcal{H}$, there exists a constant M such that $\|W_i^{[t]}\|_{\mathrm{F}} \leq M$ with probability one for all $i = 1, \ldots, L$, $t \in \mathbb{N}_+$. Furthermore, $\|F_i^{[t]}\|_{\mathrm{F}} \leq \max_{i=0,\ldots,L-1} \sqrt{d_i d_{i+1}}$ with probability one for all $i = 1, \ldots, L$, $t \in \mathbb{N}_+$. These two bounds, together with (86) and the fact that P_1, P_2 are polynomials, as well as the hypothesis that $\mathbb{E}[\|Y\|_2^m \|X\|_2^n] < \infty \, \forall m \in \{0, 1, 2\}, n \in \mathbb{N}_0$, implies the result. ■

D.1.2 Conditional expectation of $\Delta^{[t+1]}$ – Proof of Lemma 18

Let $i \in \{1, \ldots, L\}$, $r \in \{1, \ldots, d_{i+1}\}$ and $l \in \{1, \ldots, d_i\}$. Recall that $\mathcal{F}_t$ is the smallest σ-algebra generated by $\{W^{[0]}, (F^{[s]}, X^{[s]}, Y^{[s]})\}_{s \leq t}$, and note that $W^{[t]}$ is $\mathcal{F}_t$-measurable. The (i) $\mathcal{F}_t$-measurability of $W^{[t]}$ together with the (ii) hypothesis that the sequences of random variables $\{(F^{[s]}, X^{[s]}, Y^{[s]})\}_{s \in \mathbb{N}_+}$ is i.i.d. implies that

$$
\begin{aligned}
\mathbb{E}[\Delta_{i,r,l}^{[t]} | \mathcal{F}_t] &\overset{(4)}{=} \mathbb{E}\Big[\big(F_{i,r,l}^{[t+1]} \mathrm{B}_{F^{[t+1]} \odot W^{[t]}}(X^{[t+1]}, Y^{[t+1]})\big)_{i,r,l} \Big| \mathcal{F}_t \Big] \\
&\overset{\text{(i,ii)}}{=} \int F_{i,r,l} \mathrm{B}_{F \odot W^{[t]}}(X, Y)_{i,r,l} \, \mathrm{d}\mathbb{P}[F^{[t+1]} = F, X^{[t+1]} = X, Y^{[t+1]} = Y] \\
&\overset{(14)}{=} \int \Big(F_{i,r,l} \frac{\partial l(\Psi_{F \odot V^{[t]}}(X), Y)}{\partial (F_{i,r,l} V_{i,r,l})} \Big)(W^{[t]}) \, \mathrm{d}\mathbb{P}[F^{[t+1]} = F, X^{[t+1]} = X, Y^{[t+1]} = Y]
\end{aligned}
$$

$$= \int \frac{\partial l(\Psi_{F\odot V^{[t]}}(X), Y)}{\partial V_{i,r,l}}(W^{[t]})\, \mathrm{d}\mathbb{P}[F^{[t+1]} = F, X^{[t+1]} = X, Y^{[t+1]} = Y]. \qquad (87)$$

Next, we need to check that we can exchange the derivative and expectation. Note that we have the same assumptions $\mathbb{E}[\|Y\|_2^m \|X\|_2^n] < \infty \, \forall m \in \{0, 1, 2\}, n \in \mathbb{N}_+$ as for Lemma 17. as well as that $\sigma \in C^r_{\mathrm{PB}}(\mathbb{R})$. Therefore, by (86) in Lemma 17 we have that $|\Delta^{[t+1]}_{i,r,l}|$ is upper bounded and moreover $\mathbb{E}[\Delta^{[t+1]}_{i,r,l}] \leq C_{\mathcal{H}}$ for some $C_{\mathcal{H}} \leq \infty$ only dependent on $\mathcal{H}$. The interchange is then warranted by the dominated convergence theorem. Hence continuing from (87), we obtain

$$\mathbb{E}[\Delta^{[t]}_{i,r,l}|\mathcal{F}_t] = \frac{\partial}{\partial W_{i,r,l}} \int l(\Psi_{F\odot W^{[t]}}(X), Y)\, \mathrm{d}\mathbb{P}[F^{[t+1]} = F, X^{[t+1]} = X, Y^{[t+1]} = Y]$$
$$\overset{(6)}{=} \frac{\partial \mathcal{D}(W^{[t]})}{\partial W_{i,r,l}}.$$

If $\sigma \in C^r_{\mathrm{PB}}(\mathbb{R})$, then for any multi-index s on the set of weights, a bound similar to (86) holds by the chain rule:

$$|\partial^s l(Y, \Psi_{W\odot F}(X))| \leq \|Y\|_{\mathrm{F}} P_{1,s}(\|W_1\|_{\mathrm{F}}, \ldots, \|W_L\|_{\mathrm{F}}, \{\|X\|_2^j\}_{j=1}^{n_{s,1}})$$
$$+ P_{2,s}(\|W_1\|_{\mathrm{F}}, \ldots, \|W_L\|_{\mathrm{F}}, \{\|X\|_2^j\}_{j=1}^{n_{s,2}}) \qquad (88)$$

where $P_{1,s}, P_{2,s}$ are polynomials and $n_{s,1}, n_{s,2}$ are the top exponents in the expansion in $\|X\|_{\mathrm{F}}$. Hence, using the assumption $\mathbb{E}[\|Y\|_2^m \|X\|_2^n] < \infty \, \forall m \in \{0, 1, 2\}, n \in \mathbb{N}_+$, we obtain for any $W \in \mathcal{K} \subset \mathcal{W}$ a compact set that $\mathbb{E}[|\partial^s l(Y, \Psi_{W\odot F}(X))|] \leq C_{\mathcal{K}}$. In particular we can apply the dominated convergence theorem and conclude $\mathcal{D}(W) \in C^{r-1}(\mathcal{W})$ with $\partial^s \mathcal{D}(W) = \mathbb{E}[\partial^s l(Y, \Psi_{W\odot F}(X))]$. ■

D.1.3 Constant $\mathcal{D}(W)$ on a critical set – Proof of Lemma 19

We use Sard's theorem (Sard, 1942) to prove Lemma 19, which gives sufficient conditions for condition (A2.6):

Proposition 21 *(Sard, 1942) Let $f : M \to N$ be a $f \in C^r$ map between manifolds with* $\dim(M) = m$, $\dim(N) = n$. *Let* $\mathrm{Crit}(f) = \{x \in M : \nabla f(x) = 0\}$ *be the set of critical points of f. If $r > m/n - 1$, then $f(\mathrm{Crit}(f))$ has measure zero.*

Proof of Lemma 19. By Lemma 18, we have $\mathcal{D}(W) \in C^r(\mathcal{W})$. By assumption (N5) we have that if $W \in \partial\mathcal{H}$ and $\mathcal{D}(W) + \pi(W) = 0$, then $\mathcal{D}(W) = 0$. Furthermore $W \in S_j$ for some j, i.e., the critical points of $\mathcal{D}(W) + \pi(W)$ are $\{W \in \mathcal{W} \mid \nabla\mathcal{D}(W) = 0\} \cap \mathcal{H}$. We apply Sard's theorem (Proposition 21) to $\mathcal{D}(W)$. We have that if $r \geq \dim(\mathcal{W})$, then $\mathcal{D}(S_i) \subseteq \mathbb{R}$ has measure zero. Since S_i is connected there is a continuous path $z_{a,b} : [0, 1] \to S_i$ joining any two points $a, b \in S_i$. By continuity of $\mathcal{D}(W)$ we must have then $\mathcal{D}(a) = \mathcal{D}(b)$, since otherwise we would have $[\mathcal{D}(a), \mathcal{D}(b)] \subseteq \mathcal{D}(S_i)$ which has positive measure in $\mathbb{R}$. Therefore $\mathcal{D}(S_i)$ must be a constant. ■

Remark that in Lemma 19 the condition $r \geq \dim(\mathcal{W})$ cannot immediately be eliminated. When $r < \dim(\mathcal{W})$, there are examples of functions which are not constant on their connected critical sets, see e.g. Hajłasz (2003).

Appendix E. Proof of Propositions 7 and 8

We use standard tools for proving convergence to an ϵ-stationary point (for a reference, see Bottou et al. (2018)). We require first the following bounds on the variance induced by dropout.

Lemma 22 *Assume that F is a random variable satisfying (Q4). If f is a vector of random variables with distribution F, then*

(i) $\mathbb{E}\Big[\|f-\mathbb{E}[f]\|_1\Big] = 2Np(1-p)$.

(ii) $\mathbb{E}\Big[\|f-\mathbb{E}[f]\|_1^2\Big] = 2N^2p(1-p)$.

Proof We prove first (i). Recall that $f \in \{0,1\}^N$. If we denote by f_i the ith entry of f, then note that from (Q4) $\mathbb{P}[f_i = 1] = p$ and so $\mathbb{E}[|f_i - \mathbb{E}[f_i]|] = \mathbb{E}[|f_i - p|] = 2p(1-p)$. From linearity (i) follows. For (ii), we have

$$\begin{aligned}\mathbb{E}\Big[\|f-\mathbb{E}[f]\|_1^2\Big] &= \sum_i \mathbb{E}\Big[|f_i-p|^2\Big] + \sum_{i\neq j}\mathbb{E}\Big[|f_i-p||f_j-p|\Big]\\ &\le 2Np(1-p) + 2N(N-1)p(1-p) = 2N^2p(1-p),\end{aligned} \tag{89}$$

where in the last inequality we have used the Cauchy–Schwartz inequality. ■

In order to prove both Propositions 7 and 8 simultaneously, we will temporarily redefine in this section D as

$$\nabla\mathsf{D}(w) = \mathbb{E}[cF \odot \nabla\mathsf{U}(W \odot cF)], \tag{90}$$

where $c > 0$ is a constant. Later on we will specify both $c = 1$ for Proposition 7 and $c = 1/p$ for Proposition 8, respectively.

Lemma 23 *Assume (Q3) and (Q4), that is, $\nabla\mathsf{U}$ is ℓ-Lipschitz and the distribution of the filters is $\{0,1\}$-valued. Then, $\nabla\mathsf{D}$ is also $c^2\ell$-Lipschitz.*

Proof Using (i) Jensen's inequality with the norm, we have for a fixed $w, s \in \mathcal{W}$ that

$$\begin{aligned}\|\nabla\mathsf{D}(w)-\nabla\mathsf{D}(s)\|_2 &= \|\mathbb{E}_f[cf\odot\nabla\mathsf{U}(w\odot cf) - cf\odot\nabla\mathsf{U}(s\odot cf)]\|_2\\ &\overset{\text{(i)}}{\le} c\mathbb{E}_f\Big[\|f\odot\nabla\mathsf{U}(w\odot cf) - f\odot\nabla\mathsf{U}(s\odot cf)\|_2\Big]\\ &\overset{\text{(ii)}}{\le} c\mathbb{E}_f\Big[\|\nabla\mathsf{U}(w\odot cf) - \nabla\mathsf{U}(s\odot cf)\|_2\Big]\\ &\overset{\text{(iii)}}{\le} c\ell\mathbb{E}_f\Big[\|w\odot cf - s\odot cf\|_2\Big]\\ &\overset{\text{(ii)}}{\le} c^2\ell\mathbb{E}_f\Big[\|w-s\|_2\Big] = c^2\ell\|w-s\|_2\end{aligned} \tag{91}$$

where we have also used (ii) the fact that for a vector u and $\{0,1\}$-valued vector f we have $\|f \odot u\|_2 \le \|u\|_2$, (iii) $\nabla \mathsf{U}$ is ℓ-Lipschitz. ■

The proof of the following lemma can be found in Appendix E.1.

Lemma 24 *Assume (Q1)–(Q4), then for any $w \in \mathcal{W}$ with $\|w\|_2 < R$, we have*

$$\mathbb{E}\Big[\|\nabla \mathsf{D}(w) - cf \odot \nabla \mathsf{U}(w \odot cf)\|_2^2\Big] \le c^2 Np(1-p)\big(4S^2 + 6cN^2(\ell^2 R^2 + 2c\ell R)\big). \tag{92}$$

We obtain in the next lemma a simple bound for the variance of the gradient that depends on the data.

Lemma 25 *Assume (Q1)–(Q4), then for any $w \in \mathcal{W}$, we have*

$$\mathbb{E}_{z,f}\Big[\|cf \odot \nabla \mathsf{U}(w \odot cf) - cf \odot \nabla r(w \odot cf, z)\|_2^2\Big] \le 4c^2 pNS^2. \tag{93}$$

Proof We use first the definition of U as an expectation. We have

$$\begin{aligned}
&\mathbb{E}_{z,f}\Big[\|cf \odot \nabla \mathsf{U}(w \odot cf) - cf \odot \nabla r(w \odot cf, z)\|_2^2\Big] \\
&\qquad = c^2 \mathbb{E}_{z,f}\Big[\|\mathbb{E}_{z_1}[f \odot \nabla r(w \odot cf, z_1)] - f \odot \nabla r(w \odot cf, z)\|_2^2\Big] \\
&\qquad \le c^2 \mathbb{E}_{z,f}\Big[\mathbb{E}_{z_1}\Big[\|f \odot (\nabla r(w \odot cf, z_1) - \nabla r(w \odot cf, z))\|_2\Big]^2\Big] \\
&\qquad \overset{\text{(i)}}{\le} c^2 \mathbb{E}_{z,f}\Big[\mathbb{E}_{z_1}\Big[2S\|f\|_2\Big]^2\Big] \\
&\qquad \overset{\text{(ii)}}{\le} c^2 \mathbb{E}_{z,f}\Big[4S^2\|f\|_2^2\Big] = 4c^2 pNS^2,
\end{aligned} \tag{94}$$

where in (i) we have used the upper bound for $\|\nabla r(w \odot cf, z)\|_2$ from (Q2) and in (ii) that since $f_i \in \{0,1\}$ for all $i \in [N]$, we have $\|f\|_2^2 = \|f\|_1$ so using linearity with (Q4) the bound follows. ■

By (Q3)–(Q4) and Lemma 23, $\nabla \mathsf{D}$ is $c^2\ell$-Lipschitz. In this case, we can then use the following common argument: if $\nabla \mathsf{D}$ is $c^2\ell$-Lipschitz then we have the inequality

$$\mathsf{D}(W^{[t+1]}) \le \mathsf{D}(W^{[t]}) + \langle \nabla \mathsf{D}(W^{[t]}), W^{[t+1]} - W^{[t]} \rangle + \frac{c^2\ell}{2}\|W^{[t+1]} - W^{[t]}\|_2^2. \tag{95}$$

We can then use the definition of $W^{[t+1]}$ to write

$$\begin{aligned}
\mathsf{D}(W^{[t+1]}) \le \mathsf{D}(W^{[t]}) &- \alpha^{\{t\}}\langle \nabla \mathsf{D}(W^{[t]}), cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]}) \rangle \\
&+ \frac{c^2\ell(\alpha^{\{t\}})^2}{2}\|cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]})\|_2^2.
\end{aligned} \tag{96}$$

Let $\mathcal{F}_t$ be the σ-algebra of $(W^{[0]}, F^{[1]}, Z^{[1]}, \dots, W^{[t]}, F^{[t]}, Z^{[t]})$. Conditional on $\mathcal{F}_t$, $F^{[t+1]} \odot \nabla r(W^{[t]} \odot F^{[t+1]}, Z^{[t+1]})$ is an unbiased estimator of $\nabla \mathsf{D}(W^{[t]})$ so that by linearity

$$\mathbb{E}\Big[\Big\langle \nabla \mathsf{D}(W^{[t]}), cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]})\Big\rangle | \mathcal{F}_t\Big] = \|\nabla \mathsf{D}(W^{[t]})\|_2^2. \tag{97}$$

Similarly to (97), we can decompose

$$\begin{aligned}
&\mathbb{E}\Big[\|cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \\
=&\mathbb{E}\Big[\|cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]}) - cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]}) \\
&\qquad\qquad\qquad\qquad + cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \\
=& \, \mathbb{E}\Big[\|cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \\
&+ \mathbb{E}\Big[\|cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]}) - cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \\
&+ 2\mathbb{E}\Big[\Big\langle cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]}) - cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]}), \\
&\qquad\qquad\qquad\qquad cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]})\Big\rangle \Big| \mathcal{F}_t\Big] \\
=& \, \mathbb{E}\Big[\|cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \\
&+ \mathbb{E}\Big[\|cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]}) - cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big],
\end{aligned} \tag{98}$$

where in the last step the cross-term vanishes since, by using the independence assumption of $Z^{[t+1]}$ and $F^{[t]}$. If we take the expectation with respect to $Z^{[t+1]}$ first, then we find

$$\mathbb{E}_{Z^{[t+1]}}[cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]}) | \mathcal{F}_t] = cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]}). \tag{99}$$

Similarly, we can add and substract $\nabla \mathsf{D}(W^{[t]})$ in the first term and repeat the argument with the definitions of $\nabla \mathsf{U}$ and $\nabla \mathsf{D}$ in (90), where we take the expectation of (98) with respect to $F^{[t+1]}$ instead. A similar cross-term vanishes. We then obtain

$$\begin{aligned}
\mathbb{E}\Big[\|cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] &\le \|\nabla \mathsf{D}(W^{[t]})\|_2^2 \\
&+ \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]}) - cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \\
&+ \mathbb{E}\Big[\|cF^{[t+1]} \odot \nabla \mathsf{U}(W^{[t]} \odot cF^{[t+1]}) - cF^{[t+1]} \odot \nabla r(W^{[t]} \odot cF^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big].
\end{aligned} \tag{100}$$

Define the constant $J_c = S^2 + \frac{3}{2} N^2 c(\ell^2 R^2 + 2c\ell R)$ depending on c. Using the bounds of Lemma 24 together with assumption (Q5) and Lemma 25 in (100) we obtain

$$\mathbb{E}\Big[\|F^{[t+1]} \odot \nabla r(W^{[t]} \odot F^{[t+1]}, Z^{[t+1]})\|_2^2 \Big| \mathcal{F}_t\Big] \le \|\nabla \mathsf{D}(W^{[t]})\|_2^2 + 4c^2 pNS^2 + 4c^2 Np(1-p) J_c. \tag{101}$$

Substitute now (97) and (101) in (96). After taking the expectation, we can use a telescopic sum in (96) with the previous bounds, which yields

$$\sum_{t=1}^{T} \alpha^{\{t\}} \Big(1 - \frac{c^2 \ell \alpha^{\{t\}}}{2}\Big) \mathbb{E}\Big[\|\nabla \mathsf{D}(W^{[t]})\|_2^2\Big] \le \mathbb{E}[\mathsf{D}(W^{[0]})] - \mathbb{E}[\mathsf{D}(W^{[T]})]$$

$$+ 2c^4\ell Np(S^2 + (1-p)J_c)\sum_{t=1}^{T}(\alpha^{\{t\}})^2. \tag{102}$$

By (Q1) we have $\mathbb{E}[\mathsf{D}(W^{[0]})] - \mathbb{E}[\mathsf{D}(W^{[T]})] \leq 2M$ independently of c. Assuming that $\alpha^{\{t\}} < \frac{1}{c^2\ell}$ for all $t \in [T]$, we then have

$$\min_{t\in[T]} \mathbb{E}\Big[\|\nabla\mathsf{D}(W^{[t]})\|_2^2\Big] \leq \frac{4M + 4\ell c^4 Np(S^2 + (1-p)J_c)\sum_{t=1}^{T}(\alpha^{\{t\}})^2}{\sum_{t=1}^{T}\alpha^{\{t\}}}. \tag{103}$$

We not proceed with proving Propositions 7 and 8.

Proof of Propositions 7 (a) and 8 : If $\alpha^{\{t\}} = \eta$ is a constant in (103), we find

$$\min_{t\in[T]} \mathbb{E}\Big[\|\nabla\mathsf{D}(W^{[t]})\|_2^2\Big] \leq \frac{4M + 4\eta^2\ell c^4 Np(S^2 + (1-p)J_c)}{T\eta}. \tag{104}$$

Minimizing the bound over η yields that the minimum occurs at $\eta^2 = M/(\ell Nc^4p(S^2 + (1-p)J_c)T)$. For this η, the bound reads

$$\min_{t\in[T]} \mathbb{E}\Big[\|\nabla\mathsf{D}(W^{[t]})\|_2^2\Big] \leq 4\sqrt{c^4p(S^2 + (1-p)J_c)}\sqrt{\frac{M\ell N}{T}}. \tag{105}$$

For proving Proposition 7 (a), set $c = 1$ in (105) as well as $J_c = J = S^2 + \frac{3}{2}N^2(\ell^2R^2 + 2\ell R)$. Note finally that the condition $\eta < 1/(c^2\ell) = 1/\ell$ is satisfied, for example, if $p > M\ell/(NS^2T)$.

For proving Propostion 8, set $c = 1/p$ in (105) as well as $J_{1/p} = S^2 + \frac{3}{2}N^2(\ell^2R^2 + 2\ell/p)/p$. Note finally that the condition $\eta < 1/(c^2\ell) = p^2/\ell$ is satisfied, for example, if $p > M\ell/(NS^2T)$.

Proof of Proposition 7 (b): Let $c = 1$ and denote $J_1 = J$ in (103). We can also set $\alpha^{\{t\}} = 1/(\ell\sqrt{t})$. It is easily verified that for $T \geq 4$:

$$\sum_{t=1}^{T}\alpha^{\{t\}} > \frac{\sqrt{T}}{\ell} \quad \text{and} \quad \sum_{t=1}^{T}(\alpha^{\{t\}})^2 < \frac{\log(T)}{\ell^2}. \tag{106}$$

Substituting these bounds in (103) yields the result. ■

E.1 Proof of Lemma 24

Noting that we have temporarily the definition $\nabla\mathsf{D}(w) = \mathbb{E}[cf \odot \nabla\mathsf{U}(w \odot cf)]$ we can write

$$\begin{aligned}
\mathbb{E}\Big[\|\nabla\mathsf{D}(w) - cf \odot \nabla\mathsf{U}(w \odot cf)\|_2^2\Big] &= \mathbb{E}_{f_1}\Big[\|\mathbb{E}_{f_2}[cf_2 \odot \nabla\mathsf{U}(w \odot cf_2) - cf_1 \odot \nabla\mathsf{U}(w \odot cf_1)]\|_2^2\Big] \\
= \mathbb{E}_{f_1}\Big[\|\mathbb{E}_{f_2}[cf_2 \odot \nabla\mathsf{U}(w \odot cf_2) - cf_2 \odot \nabla\mathsf{U}(w \odot cf_1) + & \qquad (107) \\
&+ cf_2 \odot \nabla\mathsf{U}(w \odot cf_1) - cf_1 \odot \nabla\mathsf{U}(w \odot cf_1)]\|_2^2\Big] \\
\overset{(i)}{\leq} \mathbb{E}_{f_1}\Big[\mathbb{E}_{f_2}\Big[\|cf_2 \odot (\nabla\mathsf{U}(w \odot cf_2) - \nabla\mathsf{U}(w \odot cf_1)) + c(f_2 - f_1) \odot \nabla\mathsf{U}(w \odot cf_1)\|_2\Big]^2\Big]
\end{aligned}$$

$$\overset{\text{(ii)}}{\le} \mathbb{E}_{f_1}\Big[\mathbb{E}_{f_2}\Big[\|cf_2 \odot (\nabla\mathsf{U}(w\odot cf_2) - \nabla\mathsf{U}(w\odot cf_1))\|_2 + \|c(f_2-f_1)\odot\nabla\mathsf{U}(w\odot cf_1)\|_2\Big]^2\Big],$$

where (i) we have used Jensen's inequality for a vector-valued random variable v, namely $\|\mathbb{E}[v]\|_2 \le \mathbb{E}[\|v\|_2]$, and (ii) the subadditivity of the norm $\|a+b\|_2 \le \|a\|_2 + \|b\|_2$ for any $a, b \in \mathbb{R}^N$. We now note that

$$\begin{aligned}\|cf_2 \odot (\nabla\mathsf{U}(w\odot cf_2) - \nabla\mathsf{U}(w\odot cf_1))\|_2^2 &= \sum_i c^2 f_2^i |\nabla_i\mathsf{U}(w\odot cf_2) - \nabla_i\mathsf{U}(w\odot cf_1)|^2\\ \overset{\text{(i)}}{\le} \sum_i c^2 f_2^i \ell^2 \|w\odot cf_2 - w\odot cf_1\|_2^2 &\le \sum_i c^4 f_2^i \ell^2 \|f_2 - f_1\|_2^2 \|w\|_2^2 \overset{\text{(ii)}}{\le} c^4 \|f_2\|_1 \|f_2 - f_1\|_1 \ell^2 R^2,\end{aligned} \tag{108}$$

where we have used (i) the Lipschitzness assumption of $\nabla\mathsf{U}$ from (Q3), and (ii) the facts that $\|w\|_2^2 < R^2$ and $\|f_2\|_2^2 = \|f_2\|_1$. The latter is true because for any vector f with entries $\{-1, 0, 1\}$, $\|f\|_2^2 = \|f\|_1$. We can reason similarly with $f_1 - f_2$.

Using (Q2) we can also bound

$$\|c(f_2 - f_1)\odot\nabla\mathsf{U}(w\odot f_1)\|_2^2 \le c^2\|f_2 - f_1\|_1 S^2. \tag{109}$$

Hence, we have in (107) that

$$\begin{aligned}&\mathbb{E}_{f_1}[\mathbb{E}_{f_2}[\|cf_2 \odot (\nabla\mathsf{U}(w\odot cf_2) - \nabla\mathsf{U}(w\odot cf_1))\|_2 + \|c(f_2-f_1)\odot\nabla\mathsf{U}(w\odot cf_1)\|_2]^2]\\ &\qquad\le \mathbb{E}_{f_1}[\mathbb{E}_{f_2}[c^2\|f_2\|_1^{1/2}\|f_2-f_1\|_1^{1/2}\ell R + c\|f_2 - f_1\|_1^{1/2} S]^2]\\ &\qquad\overset{\text{(i)}}{\le} \mathbb{E}_{f_1}[\mathbb{E}_{f_2}[c^2\|f_2-f_1\|_1(c\|f_2\|_1^{1/2}\ell R + S)^2]]\\ &\qquad\le \mathbb{E}_{f_1,f_2}[c^2\|f_2-f_1\|_1(c^2\|f_2\|_1\ell^2R^2 + c\|f_2\|_1^{1/2}2S\ell R + S^2)]\\ &\qquad\overset{\text{(ii)}}{\le} \mathbb{E}_{f_1,f_2}[c^2\|f_2-f_1\|_1(c\|f_2\|_1(\ell^2R^2 + 2c\ell R) + S^2)],\end{aligned} \tag{110}$$

where (i) for a random variable v we have $\mathbb{E}[v]^2 \le \mathbb{E}[v^2]$ and (ii) $\|f_2\|_1^{1/2} \le \|f_2\|_1$ since either $\|f_2\|_1 = 0$ or $\|f_2\|_1 \ge 1$. We can now add an expectation term in the norm $\|f_2 - f_1\|_1 \le \|f_2 - \mathbb{E}[f_2]\|_1 + \|f_1 - \mathbb{E}[f_1]\|_1$ and $\|f_2\|_1 \le \|f_2 - \mathbb{E}[f_2]\|_1 + \|\mathbb{E}[f_2]\|_1$. Here, $\|\mathbb{E}[f_2]\|_1 = \|\mathbb{E}[f_1]\|_1 = pN$ by (Q4). Hence, from (110) onward we can write

$$\begin{aligned}&\mathbb{E}_{f_1,f_2}[c^2\|f_2-f_1\|_1(c\|f_2\|_1(\ell^2R^2 + 2c\ell R) + S^2)]\\ &\le \mathbb{E}_{f_1,f_2}\Big[c^2\big(\|f_2 - \mathbb{E}[f_2]\|_1 + \|f_1 - \mathbb{E}[f_1]\|_1\big)\big(c\|f_2 - \mathbb{E}[f_2]\|_1(\ell^2R^2 + 2c\ell R)(1+pN) + S^2\big)\Big]\\ &= \mathbb{E}_{f_1,f_2}\Big[c^3\big(\|f_2 - \mathbb{E}[f_2]\|_1^2 + \|f_1 - \mathbb{E}[f_1]\|_1\|f_2 - \mathbb{E}[f_2]\|_1\big)(\ell^2R^2 + 2c\ell R)(1+pN)\Big]\\ &\qquad\qquad + 2c^2S^2\mathbb{E}_{f_2}\Big[\|f_2 - \mathbb{E}[f_2]\|_1\Big]\\ &\overset{\text{(i)}}{\le} \mathbb{E}_{f_1,f_2}\Big[c^3\big(\|f_2 - \mathbb{E}[f_2]\|_1^2 + \|f_1 - \mathbb{E}[f_1]\|_1\|f_2 - \mathbb{E}[f_2]\|_1\big)(\ell^2R^2 + 2c\ell R)(1+pN)\Big]\\ &\qquad\qquad + 4c^2S^2Np(1-p)\\ &\overset{\text{(ii)}}{\le} \mathbb{E}_{f_1,f_2}\Big[c^3\big(\|f_2 - \mathbb{E}[f_2]\|_1^2 + 4N^2p^2(1-p)^2\big)(\ell^2R^2 + 2c\ell R)(1+pN)\Big] + 4c^2S^2Np(1-p)\end{aligned}$$

$$
\begin{aligned}
&\overset{\text{(iii)}}{\leq} c^3\big(2N^2p(1-p)+4N^2p^2(1-p)^2\big)\big(\ell^2R^2+2c\ell R\big)\big(1+pN\big)+4c^2S^2Np(1-p)\\
&= c^2Np(1-p)\Big(c\big(2N+4Np(1-p)\big)(1+pN)(\ell^2R^2+2c\ell R)+4S^2\Big)\\
&\overset{\text{(iv)}}{\leq} c^2Np(1-p)(4S^2+6cN^2(\ell^2R^2+2c\ell R)), \qquad (111)
\end{aligned}
$$

where we have used (i) Lemma 22(i), (ii) independence of f_1 from f_2 and Lemma 22(i) again, (iii) Lemma 22(ii), and (iv) bounded $1+pN<2N$ and $p(1-p)\leq 1/4$. ■

Appendix F. Path Representation of $\mathcal{D}(W)$ – Proofs of Lemma 10 and Corollary 11

Proof of (31). Recall that $G_F=(\mathcal{E}_F,\mathcal{V})$ is a random subgraph of $G=(\mathcal{E},\mathcal{V})$ with edge set $\mathcal{E}_F=\{e\in\mathcal{E}:F_e=1\}$. By (i) the law of total expectation, and by (ii) independence of F and (X,Y),

$$
\begin{aligned}
\mathcal{D}(W) &= \mathbb{E}\Big[\sum_{i=1}^{d_L}\big(Y_f-\sum_{\gamma\in\Gamma^i(G)}P_\gamma F_\gamma X_{\gamma_0}\big)^2\Big]\\
&\overset{\text{(i)}}{=} \sum_{g\in\mathcal{G}}\mathbb{E}\Big[\sum_{f=1}^{d_L}\big(Y_f-\sum_{\gamma\in\Gamma^f(G_F)}P_\gamma X_{\gamma_0}\big)^2\Big|\{G_F=g\}\Big]\mathbb{P}[G_F=g]\\
&\overset{\text{(ii)}}{=} \sum_{g\in\mathcal{G}}\mu_g\mathbb{E}\Big[\sum_{f=1}^{d_L}\big(Y_f-\sum_{\gamma\in\Gamma^f(g)}P_\gamma X_{\gamma_0}\big)^2\Big]. \qquad (112)
\end{aligned}
$$

Proof of (32). Expand (112) to find

$$
\mathcal{D}(W)=\sum_{g\in\mathcal{G}}\mu_g\mathbb{E}\Big[\sum_{f=1}^{d_L}\Big(Y_f^2-2Y_f\sum_{\gamma\in\Gamma^f(g)}P_\gamma X_{\gamma_0}+\sum_{\gamma\in\Gamma^f(g)}\sum_{\delta\in\Gamma^f(g)}P_\gamma X_{\gamma_0}P_\delta X_{\delta_0}\Big)\Big]. \qquad (113)
$$

Setting $\eta_\gamma=\sum_{\{g\in\mathcal{G}|\gamma\in\Gamma(g)\}}\mu_g$, we obtain

$$
\begin{aligned}
\mathcal{D}(W) &= \sum_{g\in\mathcal{G}}\mu_g\mathbb{E}\Big[\Big(\sum_{f=1}^{d_L}\sum_{\gamma\in\Gamma^f(g)}\Big(\frac{Y_f^2}{|\Gamma^f(g)|}-2Y_fP_\gamma X_{\gamma_0}\Big)+\sum_{\gamma\in\Gamma(g)}\sum_{\delta\in\Gamma^{\gamma_L}(g)}P_\gamma X_{\gamma_0}P_\delta X_{\delta_0}\Big)\Big] \qquad (114)\\
&= \sum_{\gamma\in\Gamma(G)}\eta_\gamma\mathbb{E}\Big[\big(Y_{\gamma_L}-P_\gamma X_{\gamma_0}\big)^2\Big]\\
&\quad -\sum_{g\in\mathcal{G}}\mu_g\mathbb{E}\Big[\sum_{f=1}^{d_L}\sum_{\gamma\in\Gamma^f(g)}\Big(\Big(1-\frac{1}{|\Gamma^f(g)|}\Big)Y_f^2-P_\gamma X_{\gamma_0}\sum_{\delta\in\Gamma^f(g)\setminus\{\gamma\}}P_\delta X_{\delta_0}\Big)\Big]
\end{aligned}
$$

after rearranging terms. This completes Lemma 10's proof after identifying $\mathcal{J}(W)$ and $\mathcal{R}(W)$ here as the left and right sum, respectively.

To prove Corollary 11, consider that since for an arborescence $\mathcal{R}(W) = 0$, we can write

$$\begin{aligned}
&\sum_{\gamma\in\Gamma(G)} \eta_\gamma \mathbb{E}\Big[\big(Y_{\gamma_L} - P_\gamma X_{\gamma_0}\big)^2\Big] \qquad (115)\\
&= \sum_{\gamma\in\Gamma(G)} \eta_\gamma \mathbb{E}[X_{\gamma_0}^2]\Big(\frac{\mathbb{E}[Y_{\gamma_L}X_{\gamma_0}]}{\mathbb{E}[X_{\gamma_0}^2]} - P_\gamma\Big)^2 + \sum_{\gamma\in\Gamma(G)} \eta_\gamma \Big(\mathbb{E}[Y_{\gamma_L}^2] - \frac{\mathbb{E}[Y_{\gamma_L}X_{\gamma_0}]^2}{\mathbb{E}[X_{\gamma_0}^2]}\Big)\\
&\overset{\text{(iii)}}{=} \mathcal{I}(W) + \mathcal{D}(W^{\text{opt}}).
\end{aligned}$$

Here, (iii) follows because since $\mathcal{I}(W) \geq 0$ and $\mathcal{I}(W) = 0$ at $z_\gamma = P_\gamma$, what remains must be the optimum. This completes the proofs of Lemma 10 and Corollary 11. ■

Appendix G. Conserved Quantities – Proof of Lemma 12

For any edge $f \in \mathcal{E}$,

$$\begin{aligned}
W_f \frac{\partial \mathcal{D}}{\partial W_f} &\overset{(31)}{=} \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{e=1}^{d} 2\big(Y_e - \sum_{\gamma\in\Gamma^e(g)} P_\gamma X_{\gamma_0}\big)\big(\sum_{\delta\in\Gamma^e(g;f)} P_\delta X_{\delta_0}\big)\Big]\\
&= \sum_{g\in\mathcal{G}} \mu_g \mathbb{E}\Big[\sum_{\delta\in\Gamma(g;f)} 2\big(Y_{\delta_L} - \sum_{\gamma\in\Gamma^{\delta_L}(g)} P_\gamma X_{\gamma_0}\big) P_\delta X_{\delta_0}\Big]. \qquad (116)
\end{aligned}$$

Note that $\Gamma(g;l) = \Gamma^l(g)$ for any leaf $l \in \mathcal{L}(G)$ and $g \in \mathcal{G}$, and therefore in particular

$$W_l \frac{\partial \mathcal{D}}{\partial W_l} = \sum_{g\in\mathcal{G}} \mu_g \sum_{\delta\in\Gamma^l(g)} \mathbb{E}\Big[2\big(Y_{\delta_L} - \sum_{\gamma\in\Gamma^{\delta_L}(g)} P_\gamma X_{\gamma_0}\big) P_\delta X_{\delta_L}\Big]. \qquad (117)$$

Recall that $\mathcal{L}(G; f)$ is the set of leaves of the subtree of the base graph G rooted at $f \in \mathcal{E}$. By the fact that $\{\Gamma^l(g; f)\}_{l\in\mathcal{L}(G;f)}$ partitions $\Gamma(g; f)$ for any $g \in \mathcal{G}$, viz.,

$$\Gamma(g; f) = \cup_{l\in\mathcal{L}(G;f)} \Gamma^l(g; f), \quad \Gamma^{l_1}(g; f) \cap \Gamma^{l_2}(g; f) = \emptyset \text{ for all } l_1 \neq l_2,\, g \in \mathcal{G}, \qquad (118)$$

it follows that

$$\sum_{l\in\mathcal{L}(G;f)} W_l \frac{\partial \mathcal{D}}{\partial W_l} = W_f \frac{\partial \mathcal{D}}{\partial W_f}. \qquad (119)$$

Note in fact that this proof works for *any* base graph G that has no cycles and only length-L paths, so not just an arborescence. This is why we make Assumption (N6') as opposed to the stronger Assumption (N6) in Corollary 11. ■

Appendix H. Proof of Proposition 13

The proof of Proposition 13 is by double induction on the statements $A(t) \equiv \{\mathcal{I}(W^{\{s\}}) \leq \mathcal{I}(W^{\{s-1\}})\mathrm{e}^{-2\nu_{\min}\kappa\alpha}, \forall s \in [t]\}$ and $B(t) \equiv \{W^{\{s\}} \in K, \forall s \in [t]\}$ where $\kappa > 0$ is a free parameter and K is a compact set which we will define. Concretely, we prove that there exist α and κ such that $A(t) \cap B(t) \Rightarrow B(t+1)$ and $A(t) \cap B(t+1) \Rightarrow A(t+1)$. Appendix H.4

describes in detail how the upcoming Lemmas 26–28 provide sufficient conditions for the induction step. There we also maximize the upper bound on the convergence rate over κ, which gives the rate in (41).

We start by giving Lemmas 26–28. Recall first the definition of the set $B(\epsilon, I)$ in (39). Here, with a minor abuse of notation, we define also

$$B(\epsilon, \{C_f\}_{f\in\mathcal{E}\backslash\mathcal{L}(G)}) \triangleq \big\{W \in \mathcal{W} \big| \mathcal{I}(W) \leq \epsilon, W_f^2 - \sum_{l\in\mathcal{L}(G;f)} W_{\gamma^l}^2 = C_f\big\} \tag{120}$$

where $\{\gamma^l\} \triangleq \Gamma^l(G)$ for $l \in \mathcal{L}(G)$ if G is an arborescence.

Lemma 26 *Assume (N2) from Proposition 6 and (N6) from Corollary 11. Then:*

(i) If $\epsilon > 0$ and $|C_f| < \infty$ for $f \in \mathcal{E}\backslash\mathcal{L}(G)$, then the set $B(\epsilon, \{C_f\}_{f\in\mathcal{E}\backslash\mathcal{L}})$ is compact.

(ii) If $\max_{\gamma\in\Gamma(G)} |z_\gamma| \leq M^L$, then the function $\mathcal{I}(W)$ is β-smooth in $\mathcal{S}$ with $\beta = 6\nu_{\max} \cdot |\mathcal{E}(G)|\,|\Gamma(G)|\,M^{2(L-1)}$.

Lemma 26 implies that $B(\epsilon, I)$ is compact and that $\mathcal{D}(W)$ is β-smooth on the compact set $K = \mathcal{S} \cap B(\epsilon, I)$, i.e.,

$$\mathcal{D}(W') - \mathcal{D}(W) \leq \nabla\mathcal{D}(W)^{\mathrm{T}}(W' - W) + \beta\|W' - W\|_2^2 \tag{121}$$

for $W, W' \in K$. Its proof is deferred to Appendix H.1.

Next, Lemma 27 gives a lower bound on the curvature of $\mathcal{D}(W)$ on K in the direction of $\nabla\mathcal{D}(W)$, in the form of a PL-inequality (Karimi et al., 2016). Its proof is in Appendix H.2.

Lemma 27 *Assume (N2) from Proposition 6 and (N6) from Corollary 11. If $W^{\{t\}} \in \mathcal{S} \cap B(\epsilon, I)$, then*

$$\|\nabla\mathcal{D}(W^{\{t\}})\|_2^2 \geq 4\nu_{\min}(C_{\min}^{\{t\}})^{(L-1)}\big(\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\mathrm{opt}})\big). \tag{122}$$

Lemma 28 proves that the conserved quantities of the gradient flow remain bounded under the GD algorithm in (27). This lemma allows us to keep track of the iterates in the compact set $K = \mathcal{S} \cap B(\epsilon, I)$ by relating them to conserved quantities and exploiting the fact that under GD $|C_f^{\{t+1\}} - C_f^{\{t\}}|$ has order $O(\alpha^2)$. Appendix H.2 contains its proof.

Lemma 28 *Assume (N2) from Proposition 6 and (N6) from Corollary 11. If $W^{\{t\}} \in \mathcal{S}$, and $C_f^{\{t\}} > 0$ for all $f \in \mathcal{E}\backslash\mathcal{L}(G)$, then $4\alpha^2\,\|\nu\|_1\,M^{2(L-1)}\big(\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\mathrm{opt}})\big) \geq |C_f^{\{t+1\}} - C_f^{\{t\}}|$.*

A note on the exchange of derivative and expectation in this section. Whenever we make both Assumption (N2) in Proposition 6 and (N7) in Lemma 10, the exchange of derivative and expectation is warranted. This occurs several times throughout this section. We refer to the proof of Lemma 18 for the details.

H.1 Compactness, and Smoothness – Proof of Lemma 26

In the proof of Lemma 26, we will upper bound the operator norm of the Hessian. Recall that for a symmetric bilinear matrix A, $\|A\|_{\text{op}} \triangleq \sup_{\|v\|_2=1} |v^T A v|$.

Proof of (i). By continuity of the conditions in (39), the set $B(\epsilon, \{C_f\}_{f\in\mathcal{E}\backslash\mathcal{L}})$ is closed. We need to prove boundedness. Let $W \in B(\epsilon, \{C_f\}_{f\in\mathcal{E}\backslash\mathcal{L}})$, and suppose w.l.o.g. that for some $f^* \in \mathcal{E}\backslash\mathcal{L}$ we have $|W_{f^*}| > Q$, where $Q > \max_{j\in\mathcal{E}\backslash\mathcal{L},\gamma\in\Gamma(G)}\{|C_j|, |z_\gamma|\}$. We want to find a path $\gamma \in \Gamma(G)$ such that P_γ is large for a contradiction with the assumption that $\mathcal{I}(W) \leq \epsilon$. By (35), we have the inequality $\sum_{l\in\mathcal{L}(G;f^*)} W_l^2 > Q^2 - |C_{f^*}|$ so that for some $l^* \in \mathcal{L}(G; f^*)$ we must have $W_{l^*}^2 > (Q - |C_{f^*}|)/|\mathcal{L}(G; f^*)|$. Consequently, we have by (35) that $|W_e|^2 > (Q^2 - |C_{f^*}|)/|\mathcal{L}(G; f^*)| - |C_e|$ for any edge $e \in \gamma$ in any path $\gamma \in \Gamma^{l^*}(G)$ except for the edge f^* where we have $|W_{f^*}| > Q$ by assumption. In particular, we have the bound $|W_e| > O(Q)$ for any edge $e \in \gamma$ for any path $\gamma \in \Gamma(G; f^*)$. Therefore if we pick $\gamma \in \Gamma(G; f^*)$ we have

$$\epsilon \overset{(39)}{\geq} \mathcal{I}(W) \geq \nu_\gamma(z_\gamma - P_\gamma)^2 \geq \nu_\gamma(|P_\gamma| - |z_\gamma|)^2 > O(Q^{2L}) \tag{123}$$

for sufficiently large Q, which is a contradiction. We must thus have $|W_{f^*}| \leq Q$ for some $Q < \infty$. If on the other hand $|W_l| > Q$ for some $l \in \mathcal{L}(G; f^*)$, by (35) we must also have $(W_{f^*})^2 > Q^2 + C_{f^*} > O(Q^2)$ for sufficiently large Q. This case is, thus, the same as before.
Proof of (ii). Using a regular upper bound to the entries of $\nabla^2\mathcal{I}(W)$ when $W \in \mathcal{S}$ will suffice. Element-wise, we have

$$(\nabla^2\mathcal{I}(W))_{i,j} \tag{124}$$
$$= \begin{cases} 2\sum_{\delta\in\Gamma(G;i)\cap\Gamma(G;j)} \nu_\delta\Big(\frac{P_\delta}{W_i}\frac{P_\delta}{W_j} - \frac{P_\delta}{W_iW_j}(z_\gamma - P_\gamma)\Big), & \text{if } i \neq j, \Gamma(G;i)\cap\Gamma(G;j) \neq \emptyset, \\ 2\sum_{\gamma\in\Gamma(G;i)} \nu_\gamma(\frac{P_\gamma}{W_i})^2 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, noting that since we have $|W_f| \leq M$ for all $f \in \mathcal{E}$ on $\mathcal{S}$, we can bound $|P_\gamma/W_f| \leq M^{L-1}$, $|z_\gamma| \leq M^L$ and the other terms similarly. We upper bound the number of terms in the sum over $\Gamma(G; i)$ and $\Gamma(G; i) \cap \Gamma(G; j)$ by $|\Gamma(G)|$ and $\nu_\gamma \leq \nu_{\max}$. Adding all terms, we obtain that $6\nu_{\max}|\Gamma(G)|\, M^{2(L-1)}$ is an upper bound for each of the entries of $\nabla^2\mathcal{I}(W)$. This gives an upper bound $\|\nabla^2\mathcal{I}(W)\|_{\text{op}} \leq 6|\mathcal{E}|\nu_{\max}|\Gamma(G)|\, M^{2(L-1)}$ in $\mathcal{S}$. ■

H.2 PL-inequality on a Compact Set – Proof of Lemma 27

Recall the definition of a PL-inequality:

Definition 29 *Let $u \in C^2(K, \mathbb{R})$ where $K \subset \mathbb{R}^n$ is compact and $K\backslash\partial K \neq \emptyset$. Denote by $u^* = \min_{x\in K} u(x)$ and suppose that $u^* \in K\backslash\partial K$. We say that u satisfies a* Polyak–Łojasiewicz (PL) *inequality if there exist a $\tau_K > 0$ depending only on K such that*

$$\|\nabla u(x)\|_2^2 \geq \tau_K(u(x) - u^*) \quad \textit{for all} \quad x \in K. \tag{125}$$

A PL-inequality together with β-smoothness on a compact set will imply that $\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\text{opt}})$ decreases. To see this, note that by (i) β-smoothness, and (ii) the update rule

$$\mathcal{D}(W^{\{t+1\}}) - \mathcal{D}(W^{\{t\}}) \overset{\text{(i)}}{\leq} \nabla\mathcal{D}(W^{\{t\}})^{\mathrm{T}}(W^{\{t+1\}} - W^{\{t\}}) + \beta\|W^{\{t+1\}} - W^{\{t\}}\|_2^2$$

$$\overset{\text{(ii)}}{=} \alpha\big(\beta\alpha - 1\big)\|\nabla\mathcal{D}(W^{\{t\}})\|_2^2 \tag{126}$$

If furthermore $\alpha \leq 1/(2\beta)$, then also $\beta\alpha - 1 \leq -1/2$. Together with (125), and after rearranging terms, one finds that

$$\mathcal{D}(W^{\{t+1\}}) - \mathcal{D}(W^{\{t\}}) \leq \frac{\alpha\tau_K}{2}(\mathcal{D}(W^{\{t\}}) - \mathcal{D}(W^{\text{opt}})) \quad \text{for all} \quad W \in K. \tag{127}$$

By (iii) $1 + x \leq \mathrm{e}^x$ for all $x \in \mathbb{R}$, we obtain (29). The strategy will now be to prove that there is a PL-inequality in some compact set, that the iterates remain in that compact set, and that the function is β-smooth.

Proof of Lemma 27. First note that if $l \in \mathcal{L}(G)$ and $\gamma \in \Gamma(G; l)$, the indexes of the weights in the product $|P_\gamma^{\{t\}}/W_l^{\{t\}}|$ belong to the index set $\mathcal{E}\backslash\mathcal{L}(G)$. The proof follows (i) by restricting the sum, and (ii) from the fact that for every path $\gamma \in \Gamma(G)$ in an arborescence G, there is exactly one leaf $l \in \mathcal{L}(G)$ such that $\gamma^l = \gamma$. Thus

$$\begin{aligned}\sum_{e\in\mathcal{E}}\Big|\frac{\partial}{\partial W_e}\mathcal{I}(W^{\{t\}})\Big|^2 &= 4\sum_{e\in\mathcal{E}}\Big|\sum_{\gamma\in\Gamma(G;e)}\nu_\gamma\frac{P_\gamma^{\{t\}}}{W_e^{\{t\}}}(z_\gamma - P_\gamma^{\{t\}})\Big|^2 \overset{\text{(i)}}{\geq} 4\sum_{l\in\mathcal{L}(G)}\Big|\nu_{\gamma^l}\frac{P_{\gamma^l}^{\{t\}}}{W_l^{\{t\}}}(z_{\gamma^l} - P_{\gamma^l}^{\{t\}})\Big|^2 \\ &\overset{\text{(ii)}}{=} 4\sum_{\gamma\in\Gamma(G)}\nu_\gamma^2\Big|\frac{P_\gamma^{\{t\}}}{W_{\gamma_L}^{\{t\}}}(z_\gamma - P_\gamma^{\{t\}})\Big|^2 \overset{\text{(iii)}}{\geq} 4\nu_{\min}\big(\min_{f\in\mathcal{E}\backslash\mathcal{L}(G)}|W_f^{\{t\}}|^2\big)^{L-1}\mathcal{I}(W^{\{t\}}),\end{aligned} \tag{128}$$

where in (iii) we have used the bound $|W_i^{\{t\}}| \geq \min_{e\in\mathcal{E}\backslash\mathcal{L}(G)}|W_e^{\{t\}}|$ for all $i \in \mathcal{E}\backslash\mathcal{L}(G)$ and similarly with $\nu_\gamma \geq \nu_{\min}$ for $\gamma \in \Gamma(G)$. Finally, by (35), we have $\min_{e\in\mathcal{E}\backslash\mathcal{L}(G)}|W_e^{\{t\}}|^2 \geq C_{\min}^{\{t\}}$. This completes the proof. ■

H.3 Conserved Quantities Remain Bounded throughout GD – Proof of Lemma 28

Proof Pick $f \in \mathcal{E}\backslash\mathcal{L}(G)$. By (i) Corollary 11, and (ii) Lemma 12, we have

$$\begin{aligned}C_f^{\{t+1\}} &= (W_f^{\{t+1\}})^2 - \sum_{l\in\mathcal{L}(G;i)}(W_l^{\{t+1\}})^2 \\ &\overset{(27)}{=} \Big(W_f^{\{t\}} - \alpha\frac{\partial}{\partial W_f}\mathcal{D}(W^{\{t\}})\Big)^2 - \sum_{l\in\mathcal{L}(G;f)}\Big(W_l^{\{t\}} - \alpha\frac{\partial}{\partial W_l}\mathcal{D}(W^{\{t\}})\Big)^2 \\ &\overset{\text{(i)}}{=} \Big(W_f^{\{t\}} - \alpha\frac{\partial}{\partial W_f}\mathcal{I}(W^{\{t\}})\Big)^2 - \sum_{l\in\mathcal{L}(G;f)}\Big(W_l^{\{t\}} - \alpha\frac{\partial}{\partial W_l}\mathcal{I}(W^{\{t\}})\Big)^2 \\ &\overset{\text{(ii)}}{=} C_f^{\{t\}} + \alpha^2\Big(\Big(\frac{\partial}{\partial W_f}\mathcal{I}(W^{\{t\}})\Big)^2 - \sum_{l\in\mathcal{L}(G;f)}\Big(\frac{\partial}{\partial W_l}\mathcal{I}(W^{\{t\}})\Big)^2\Big) \\ &= C_i^{\{t\}} + 4\alpha^2\Big(\Big(\sum_{\gamma\in\Gamma(G;f)}\nu_\gamma\frac{P_\gamma^{\{t\}}}{W_f^{\{t\}}}(z_\gamma - P_\gamma^{\{t\}})\Big)^2 - \sum_{l\in\mathcal{L}(G;f)}\nu_{\gamma^l}^2\Big(\frac{P_{\gamma^l}^{\{t\}}}{W_l^{\{t\}}}\Big)^2(z_{\gamma^l} - P_{\gamma^l}^{\{t\}})^2\Big) \qquad (129) \\ &\geq C_f^{\{t\}} - 4\alpha^2\Big(\sum_{l\in\mathcal{L}(G;f)}\nu_{\gamma^l}^2\Big(\frac{P_{\gamma^l}^{\{t\}}}{W_l^{\{t\}}}\Big)^2(z_{\gamma^l} - P_{\gamma^l}^{\{t\}})^2\Big). \qquad (130)\end{aligned}$$

By Cauchy–Schwartz we also have

$$\Big(\sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \frac{P_\gamma^{\{t\}}}{W_f^{\{t\}}} (z_\gamma - P_\gamma^{\{t\}}) \Big)^2 \leq \Big(\sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \Big) \sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \Big(\frac{P_\gamma^{\{t\}}}{W_l^{\{t\}}} \Big)^2 (z_\gamma - P_\gamma^{\{t\}})^2. \tag{131}$$

If we have $C_f^{\{t\}} > 0$, then $(W_f^{\{t\}})^2 > (W_{\gamma_L}^{\{t\}})^2$ for any $\gamma \in \Gamma(G;f)$. Thus, combining the estimate (129) with (131) we obtain

$$C_f^{\{t+1\}} \leq C_f^{\{t\}} + 4 \Big(\sum_{\gamma \in \Gamma(G;f)} \nu_\gamma \Big) \alpha^2 \Big(\sum_{l \in \mathcal{L}(G;f)} \nu_{\gamma^l} \Big(\frac{P_{\gamma^l}^{\{t\}}}{W_l^{\{t\}}} \Big)^2 (z_{\gamma^l} - P_{\gamma^l}^{\{t\}})^2 \Big). \tag{132}$$

Extending the sums in (132) from $\Gamma(G;f)$ to $\Gamma(G)$ and from $\mathcal{L}(G;f)$ to $\mathcal{L}(G)$, respectively, yields

$$C_f^{\{t+1\}} - C_f^{\{t\}} \leq 4 \left\| \nu \right\|_1 \alpha^2 \big(\max_{e \in \mathcal{E} \setminus \mathcal{L}(G)} |W_e^{\{t\}}|^2 \big)^{L-1} \mathcal{I}(W^{\{t\}}), \tag{133}$$

where we have used the bound $|W_f| \leq \max_{e \in \mathcal{E} \setminus \mathcal{L}(G)} |W_e|$ for all $f \in \mathcal{E} \setminus \mathcal{L}(G)$. Similarly, using (130) and the trivial bound $\nu_\gamma \leq \left\| \nu \right\|_1$ for any $\gamma \in \Gamma$, and by absorbing one ν_γ-term into $\mathcal{I}(W)$'s expression, we obtain

$$C_f^{\{t+1\}} \geq C_f^{\{t\}} - 4 \left\| \nu \right\|_1 \alpha^2 \big(\max_{e \in \mathcal{E} \setminus \mathcal{L}(G)} |W_e^{\{t\}}|^2 \big)^{L-1} \mathcal{I}(W^{\{t\}}) \tag{134}$$

for the lower bound. Because $W^{\{t\}} \in \mathcal{S}$ by assumption, $\max_{e \in \mathcal{E} \setminus \mathcal{L}(G)} |W_e^{\{t\}}|^2 \leq M^2$. This completes the proof. ■

H.4 Double Induction

We now use Lemmas 26–28 together in a double induction to finally prove Proposition 13. Let $\kappa > 0$ and denote the statements:

$$A(t) \equiv \{ \mathcal{I}(W^{\{s\}}) \leq \mathcal{I}(W^{\{s-1\}}) \mathrm{e}^{-2\nu_{\min}\kappa\alpha}, \forall s \in [t] \}, \tag{135}$$

$$B(t) \equiv \{ W^{\{s\}} \in B(\epsilon, I) \cap \mathcal{S} \, \forall s \in [t] \}. \tag{136}$$

We will prove that there exists a $\kappa > 0$ such that when choosing α appropriately, firstly

$$A(t) \cap B(t) \Rightarrow B(t+1), \tag{137}$$

and secondly,

$$A(t) \cap B(t+1) \Rightarrow A(t+1). \tag{138}$$

Step 1: $A(t) \cap B(t) \Rightarrow B(t+1)$. We need to prove that $W^{\{t+1\}} \in B(\epsilon, I) \cap \mathcal{S}$ assuming (135) and (136). Using (133) from the proof of Lemma 28 repeatedly with the bound $\max_{e \in \mathcal{E}} |W_e^{\{t\}}| \leq M$, we obtain

$$C_f^{\{t+1\}} \leq C_f^{\{0\}} + 4 \left\| \nu \right\|_1 M^{2(L-1)} \alpha^2 \sum_{s=0}^{t} \mathcal{I}(W^{\{s\}}). \tag{139}$$

By (135), we can upper bound

$$\sum_{s=0}^{t} \mathcal{I}(W^{\{s\}}) \overset{(135)}{\leq} \sum_{s=0}^{t} \mathcal{I}(W^{\{0\}}) \exp(-2\nu_{\min}\kappa\alpha s) \leq \mathcal{I}(W^{\{0\}}) \frac{1}{1-\mathrm{e}^{-2\nu_{\min}\kappa\alpha}}. \tag{140}$$

If furthermore (C1) $0 < 2\nu_{\min}\kappa\alpha < 1$, then (i) the inequality $1/(1-\exp(-2\nu_{\min}\kappa\alpha)) < 1/(\nu_{\min}\kappa\alpha)$ holds, so that

$$C_{\min}^{\{t+1\}} \overset{(139)}{\leq} C_{\min}^{\{0\}} + 4\left\|\nu\right\|_1 M^{2(L-1)}\alpha^2 \sum_{s=0}^{t} \mathcal{I}(W^{\{s\}}) \overset{\text{(i)}}{\leq} C_{\min}^{\{0\}} + 4\frac{\|\nu\|_1}{\nu_{\min}} M^{L-1}\alpha\kappa^{-1}\mathcal{I}(W^{\{0\}}). \tag{141}$$

In the same manner, we can also prove (141) for $C_f^{\{0\}}$ instead of $C_{\min}^{\{0\}}$. This yields

$$C_f^{\{t+1\}} \leq C_f^{\{0\}} + 4\frac{\|\nu\|_1}{\nu_{\min}\kappa} M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}}) \tag{142}$$

for any $f \in \mathcal{E}\backslash\mathcal{L}(G)$. Similarly, for a lower bound, we can use (134) repeatedly together with the bound (140) and condition (C1) yielding

$$C_f^{\{t+1\}} \geq C_f^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}\kappa} M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}}). \tag{143}$$

for any $f \in \mathcal{E}\backslash\mathcal{L}(G)$. Now, suppose (D1) $C_{\min}^{\{0\}} - \kappa^{1/(L-1)} > 0$ and let (C2) the step size satisfy

$$\alpha \leq \nu_{\min}\kappa \frac{C_{\min}^{\{0\}} - \kappa^{1/(L-1)}}{8\left\|\nu\right\|_1 M^{2(L-1)}\mathcal{I}(W^{\{0\}})}. \tag{144}$$

We have (i) by (142) and (143) that

$$\begin{aligned} C_f^{\{t+1\}} &\overset{\text{(i)}}{\in} [C_f^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}} M^{2(L-1)}\alpha\kappa^{-1}\mathcal{I}(W^{\{0\}}), C_f^{\{0\}} + 4\frac{\|\nu\|_1}{\nu_{\min}} M^{2(L-1)}\alpha\kappa^{-1}\mathcal{I}(W^{\{0\}})] \\ &\overset{(144)}{\subseteq} [C_f^{\{0\}} - \frac{1}{2}(C_{\min}^{\{0\}} - \kappa^{1/(L-1)}), C_f^{\{0\}} + \frac{1}{2}(C_{\min}^{\{0\}} - \kappa^{1/(L-1)})] \\ &\overset{\text{(D1)}}{\subseteq} [C_f^{\{0\}} - C_f^{\{0\}}/2, C_f^{\{0\}} + C_f^{\{0\}}/2] \subseteq [C_f^{\{0\}}/2, 3C_f^{\{0\}}/2] = I_f. \end{aligned} \tag{145}$$

Then $W^{\{t+1\}} \in B(\epsilon, I)$ by (39). Hence, $M > W_f^{\{t+1\}} \overset{(35)}{>} (C_f^{\{0\}}/2)^{1/2} \geq (C_{\min}^{\{0\}}/2)^{1/2} > \delta$ for any $f \in \mathcal{E}\backslash\mathcal{L}(G)$. Since moreover $C_e^{\{t+1\}} > 0$ for all $e \in \mathcal{E}\backslash\mathcal{L}(G)$, we have that if $f \in \mathcal{L}(G)$, then $M^2 > (W_j^{\{t+1\}})^2 > (W_f^{\{t+1\}})^2$ for some $j \in \mathcal{E}\backslash\mathcal{L}(G)$. Consequently $M \geq |W_f^{\{t+1\}}|$ and $W^{\{t+1\}} \in \mathcal{S}$.

Step 2: $A(t) \cap B(t+1) \Rightarrow A(t+1)$. Suppose that $W^{\{s\}} \in B(\epsilon, I) \cap \mathcal{S}$ for $s = 0, 1, \ldots, t+1$. Using the bound in (142) which requires the induction hypothesis $A(t)$ and (C1) for $C_{\min}^{\{t\}}$, we obtain

$$C_{\min}^{\{t\}} \geq C_{\min}^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}\kappa} M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}}). \tag{146}$$

Suppose now for a moment that (C2) the right-hand side of (146) is positive for some sufficiently small α. We could then use the PL inequality from Lemma 27 together with $\min_{e\in\mathcal{E}\setminus\mathcal{L}(G)} |W_e^{\{t\}}|^{2(L-1)} \geq (C_{\min}^{\{t\}})^{L-1}$, that is,

$$\|\nabla\mathcal{I}(W^{\{t\}})\|_2^2 \geq 4\nu_{\min}(C_{\min}^{\{t\}})^{L-1}\mathcal{I}(W^{\{t\}}). \tag{147}$$

To see how, note that the argumentation around (127) together with (147) and (i) the induction hypothesis $B(t+1)$ we have $W^{\{t\}}, W^{\{t+1\}} \in B(\epsilon, I) \cap \mathcal{S}$ and (ii) the clause (L1) $\alpha \leq 1/(2\beta)$, implies

$$\begin{aligned}
\mathcal{I}(W^{\{t+1\}}) &\overset{\text{(i,ii, 147)}}{\leq} \mathcal{I}(W^{\{t\}})\exp\bigl(-2\nu_{\min}\alpha(C_{\min}^{\{t\}})^{L-1}\bigr)\\
&\overset{(146)}{\leq} \mathcal{I}(W^{\{t\}})\exp\Bigl(-2\nu_{\min}\alpha\bigl(C_{\min}^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}\kappa}M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}})\bigr)\Bigr)\\
&\overset{\text{(iii)}}{\leq} \mathcal{I}(W^{\{0\}})\exp\Bigl(-2\nu_{\min}\alpha\bigl(C_{\min}^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}\kappa}M^{2(L-1)}\alpha\mathcal{I}(W^{\{0\}})\bigr)^{L-1} - 2\nu_{\min}\alpha\kappa t\Bigr)
\end{aligned} \tag{148}$$

where we have also used (iii) the induction hypothesis $A(t)$, i.e., that $\mathcal{I}(W^{\{t\}}) \leq \mathcal{I}(W^{\{0\}}) \cdot \mathrm{e}^{-2\nu_{\min}\kappa\alpha t}$.

We now investigate the exponent in (148) for a moment. Assuming (C2) and if (C3) the right-hand side of (148) is furthermore smaller than $\mathcal{I}(W^{\{0\}})\exp(-2\nu_{\min}\kappa\alpha(t+1))$, then the induction step would be complete. Note finally that both conditions (C2) and (C3) are satisfied when choosing

$$\kappa \leq \bigl(C_{\min}^{\{0\}} - 4\frac{\|\nu\|_1}{\nu_{\min}}M^{2(L-1)}\alpha\kappa^{-1}\mathcal{I}(W^{\{0\}})\bigr)^{L-1} \tag{149}$$

or equivalently

$$\alpha \leq \nu_{\min}\kappa\frac{C_{\min}^{\{0\}} - \kappa^{1/(L-1)}}{4\,\|\nu\|_1\,M^{2(L-1)}\mathcal{I}(W^{\{0\}})}. \tag{150}$$

To also satisfy condition (C1), we thus require that

$$\alpha \leq \min\Bigl(\frac{1}{2\nu_{\min}\kappa}, \nu_{\min}\kappa\frac{C_{\min}^{\{0\}} - \kappa^{1/(L-1)}}{4\,\|\nu\|_1\,M^{2(L-1)}\mathcal{I}(W^{\{0\}})}\Bigr). \tag{151}$$

Step 3. Let us summarize. Convergence occurs at rate at most $2\nu_{\min}\kappa\alpha$ if conditions (L1), (D1), (C1)–(C3) hold. Hence we have to choose $\kappa > 0$ such that $C_{\min}^{\{0\}} - \kappa^{L-1} > 0$ and

$$\alpha \leq \min\Bigl(\nu_{\min}\kappa\frac{C_{\min}^{\{0\}} - \kappa^{1/(L-1)}}{8\,\|\nu\|_1\,M^{2(L-1)}\mathcal{I}(W^{\{0\}})}, \frac{1}{2\beta}, \frac{1}{2\nu_{\min}\kappa}\Bigr). \tag{152}$$

Note that we can maximize the convergence rate $2\nu_{\min}\alpha\kappa$ by maximizing $\kappa^2(C_{\min}^{\{0\}} - \kappa^{1/(L-1)})$, which occurs when $\kappa = (C_{\min}^{\{0\}})^{L-1}(1 + 1/(2(L-1)))^{-(L-1)} \geq \mathrm{e}^{-1/2}(C_{\min}^{\{0\}})^{L-1}$. Substituting this in (152) we require a step size

$$\alpha \leq \min\Bigl(\nu_{\min}\frac{\mathrm{e}^{1/2}(C_{\min}^{\{0\}})^{L}}{8\,\|\nu\|_1\,(2L-1)M^{2(L-1)}\mathcal{I}(W^{\{0\}})}, \frac{1}{2\beta}, \frac{1}{2\nu_{\min}(C_{\min}^{\{0\}})^{L-1}}\Bigr). \tag{153}$$

Finally, we have the bound $\beta \leq 6\nu_{\max} |\mathcal{E}(G)| |\Gamma(G)| M^{2(L-1)}$ from Lemma 26 in $\mathcal{S}$, so that

$$\alpha \leq \min\Big(\nu_{\min} \frac{\mathrm{e}^{1/2}(C_{\min}^{\{0\}})^L}{16 \|\nu\|_1 LM^{2(L-1)}\mathcal{I}(W^{\{0\}})}, \frac{1}{12\nu_{\max} |\mathcal{E}(G)| |\Gamma(G)| M^{2(L-1)}}, \frac{1}{2\nu_{\min}(C_{\min}^{\{0\}})^{L-1}}\Big). \tag{154}$$

This completes our proof of Proposition 13. ■

Appendix I. Convergence Rate in the Case of *Dropout* and *Dropconnect* – Proof of Proposition 9

We consider first the case of *Dropconnect*. We have that $\{F_e\}_{e\in\mathcal{E}}$ are independent and identically distributed Bernoulli(p) random variables. Suppose that the base graph G has no cycles and every path is of length L. Then by definition in Lemma 10, we have

$$\begin{aligned}\eta_\gamma &= \sum_{\{g\in\mathcal{G}|\gamma\in\Gamma(g)\}} \mathbb{P}[G_F = g] = \sum_{g\in\mathcal{G}} \mathbb{1}[\gamma \in \Gamma(g)]\mathbb{P}[G_F = g] \\ &= \sum_{g\in\mathcal{G}} \mathbb{P}[\gamma\in\Gamma(g)|G_F = g]\mathbb{P}[G_F = g] = \mathbb{P}[\gamma \in \Gamma(G_F)] \overset{\text{(i)}}{=} p^L \end{aligned}\tag{155}$$

where (i) we have used *Dropconnect*'s distribution on F.

Now suppose that additionally we make the stronger assumption that G is an arborescence. Then by definition in Corollary 11 $\nu_\gamma = \mathbb{E}[X^2]\eta_\gamma$, and subsequently we can calculate $\|\nu\|_1 = \mathbb{E}[X^2]\sum_{\gamma\in\Gamma(G)}\nu_\gamma = \mathbb{E}[X^2]\,|\Gamma(G)|\,p^L = \mathbb{E}[X^2]d_Lp^L = \mathbb{E}[X^2]|\mathcal{L}(G)|p^L$.

Now, since by assumption $\max_\gamma |z_\gamma| \leq M^L$ and $|W_f| \leq M$ for all $f \in \mathcal{E}$, then $\mathcal{I}(W^{\{0\}}) \leq O(|\Gamma(G)|\, M^{2L})$ so that substitution of in the definition of α in Proposition 13 yields

$$\alpha = O\Big(\frac{(C_{\min}^{\{0\}})^L}{LM^{4L}}\Big), \tag{156}$$

where we have used that $C_{\min} \leq M^2$. Finally multiplying by τ gives the rate

$$\alpha\tau = O\Big(\frac{p^L(C_{\min}^{\{0\}})^2 L}{L|\mathcal{L}(G)|^2M^{4L}}\Big). \tag{157}$$

Substituting these results in the rate $\tau\alpha$ in Proposition 13 yields the result for *Dropconnect*.

Finally we note that for the case of *Dropout*, filtering all nodes independently in an arborescence is equivalent to filtering all edges independently except the edge at the root. In particular, in (155), we have $\mathbb{P}[\gamma\in\Gamma(G_F)] = p^{L-1}$. The remaining steps of the proof are then the same as for *Dropconnect* and comparing p^L with p^{L-1} we can absorb the missing p factor into the O notation, which does not change the order in L. ■

Appendix J. Inequalities Pertaining to the Frobenius Norm

Lemma 30 *For any matrix $A \in \mathbb{R}^{m\times n}$ and $1 \leq k < \infty$, it holds that $\sum_{i,j}(1 + A_{ij}^2)^k \leq nm(1 + \|A\|_{\mathrm{F}})^{2k}$. For any two matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$ and $0 \leq k < \infty$, it holds that $(1 + \|AB\|_{\mathrm{F}})^k \leq (1 + \|A\|_{\mathrm{F}})^k(1 + \|B\|_{\mathrm{F}})^k$. For any two matrices $A, B \in \mathbb{R}^{n\times m}$, it holds that $\|A \odot B\|_{\mathrm{F}} \leq \|A\|_{\mathrm{F}}\,\|B\|_{\mathrm{F}}$.*

Proof Recall Minkowski's inequality for sequences; that is $\big(\sum_i |x_i+y_i|^k\big)^{1/k} \leq \big(\sum_i |x_i|^k\big)^{1/k} + \big(\sum_i |y_i|^k\big)^{1/k}$, which holds for $1 \leq k < \infty$. It (i) implies that for any matrix $A \in \mathbb{R}^{n\times m}$ and $1 \leq k < \infty$, that

$$\sum_{i,j}(1+A_{ij}^2)^k \overset{\text{(i)}}{\leq} \Big((nm)^{1/k} + \big(\sum_{i,j}|A_{i,j}^2|^k\big)^{1/k}\Big)^k \overset{\text{(ii)}}{\leq} nm\Big(1+\big(\sum_{i,j}|A_{i,j}^2|^k\big)^{1/k}\Big)^k \tag{158}$$

where (ii) we have used that the function z^k is nondecreasing in $z \geq 0$ whenever $k \geq 0$. Because (iii) for the ℓ_k-norm for sequences it holds that $\|x\|_{2k}^2 \leq \|x\|_2^2$ whenever $1 \leq k < \infty$, we obtain

$$\sum_{i,j}(1+A_{ij}^2)^k \overset{\text{(iii)}}{\leq} nm(1+\|A\|_{\mathrm{F}}^2)^k \overset{\text{(iv)}}{\leq} nm(1+\|A\|_{\mathrm{F}})^{2k} \tag{159}$$

where (iv) we have used that the function $(1+z^2)^k \leq (1+z)^{2k}$ for all $z \geq 0$ whenever $k \geq 0$. This proves the first inequality.

The second inequality is an immediate consequence of the submultiplicativity property of the Frobenius norm and its positivity, i.e.,

$$1+\|AB\|_{\mathrm{F}} \leq 1+\|A\|_{\mathrm{F}}\|B\|_{\mathrm{F}} \leq 1+\|A\|_{\mathrm{F}}+\|B\|_{\mathrm{F}}+\|A\|_{\mathrm{F}}\|B\|_{\mathrm{F}}. \tag{160}$$

Raising to the k-th power left and right finishes its proof.

The third inequality follows from strict positivity of the summands:

$$\|A\odot B\|_{\mathrm{F}}^2 = \sum_{i,j} A_{ij}^2 B_{ij}^2 \leq \Big(\sum_{i,j} A_{ij}^2\Big)\Big(\sum_{i,j} B_{ij}^2\Big) = \|A\|_{\mathrm{F}}^2\|B\|_{\mathrm{F}}^2. \tag{161}$$

Each of the inequalities has now been shown. ■

References

Sanjeev Arora, Noah Golowich, Nadav Cohen, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Jimmy Ba and Brendan Frey. Adaptive Dropout for training deep neural networks. In *Advances in Neural Information Processing Systems*, pages 3084–3092, 2013.

Pierre Baldi and Peter Sadowski. The Dropout learning algorithm. *Artificial intelligence*, 210:78–122, 2014.

Pierre Baldi and Peter J. Sadowski. Understanding Dropout. In *Advances in Neural Information Processing Systems*, pages 2814–2822, 2013.

Peter L. Bartlett, David P. Helmbold, and Philip M. Long. Gradient descent with identity initialization efficiently learns positive-definite linear transformations by deep residual networks. *Neural Computation*, 31:477–502, 2018.

Dimitri P. Bertsekas and John N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pages 560–564. IEEE, 1995.

Vivek S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Jacopo Cavazza, Pietro Morerio, Benjamin Haeffele, Connor Lane, Vittorio Murino, and Rene Vidal. Dropout as a low-rank regularizer for matrix factorization. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 435–444, 2018.

Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Yoel Drori and Ohad Shamir. The complexity of finding stationary points with stochastic gradient descent. In *International Conference on Machine Learning*, pages 2658–2667. PMLR, 2020.

Tianxiang Gao, Hailiang Liu, Jia Liu, Hridesh Rajan, and Hongyang Gao. A global convergence theory for deep relu implicit networks via over-parameterization. In *International Conference on Learning Representations*, 2021.

Piotr Hajłasz. Whitney's example by way of Assouad's embedding. *Proceedings of the American Mathematical Society*, 131(11):3463–3467, 2003.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Wei Huang, Richard Yi Da Xu, Weitao Du, Yutian Zeng, and Yunce Zhao. Mean field theory for deep dropout networks: digging up gradient backpropagation deeply. *arXiv preprint arXiv:1912.09132*, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal–gradient methods under the Polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

Edmund Kay and Anurag Agarwal. Dropconnected neural network trained with diverse features for classifying heart sounds. In *2016 Computing in Cardiology Conference (CinC)*, pages 617–620. IEEE, 2016.

Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Durk P. Kingma, Tim Salimans, and Max Welling. Variational Dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

Harold Kushner and G. George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Zhe Li, Boqing Gong, and Tianbao Yang. Improved Dropout for shallow and deep learning. In *Advances in Neural Information Processing Systems*, pages 2523–2531, 2016.

Poorya Mianjy and Raman Arora. On Dropout and nuclear norm regularization. In *International Conference on Machine Learning*, pages 4575–4584, 2019.

Poorya Mianjy and Raman Arora. On convergence and generalization of dropout training. *Advances in Neural Information Processing Systems*, 33, 2020.

Poorya Mianjy, Raman Arora, and Rene Vidal. On the implicit bias of Dropout. In *International Conference on Machine Learning*, pages 3540–3548, 2018.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational Dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pages 2498–2507, 2017.

Anthony P. Morse. The behavior of a function on its critical set. *Annals of Mathematics*, pages 62–70, 1939.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

Samet Oymak. Learning compact neural networks with regularization. In *International Conference on Machine Learning*, pages 3966–3975, 2018.

Ambar Pal, Connor Lane, René Vidal, and Benjamin D. Haeffele. On the regularization properties of structured dropout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7671–7679, 2020.

Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 285–290. IEEE, 2014.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

Arthur Sard. The measure of the critical values of differentiable maps. *Bulletin of the American Mathematical Society*, 48(12):883–890, 1942.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. Recurrent dropout without memory loss. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1757–1766, 2016.

Albert Senen-Cerda and Jaron Sanders. Asymptotic convergence rate of dropout on shallow linear neural networks. In *Abstract Proceedings of the 2022 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, pages 105–106, 2022.

Ohad Shamir. Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. In *Conference on Learning Theory*, pages 2691–2713, 2019.

Joachim Sicking, Maram Akila, Tim Wirtz, Sebastian Houben, and Asja Fischer. Characteristics of Monte Carlo dropout in wide neural networks. *arXiv preprint arXiv:2007.05434*, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Salma Tarmoun, Guilherme Franca, Benjamin D Haeffele, and Rene Vidal. Understanding the dynamics of gradient flow in overparameterized linear models. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10153–10161, 18–24 Jul 2021.

Gregor Urban, Kevin Bache, Duc T.T. Phan, Agua Sobrino, Alexander K. Shmakov, Stephanie J. Hachey, Christopher C.W. Hughes, and Pierre Baldi. Deep learning for drug discovery and cancer research: Automated analysis of vascularization images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(3):1029–1035, 2018.

Stefan Wager, Sida Wang, and Percy S. Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 351–359, 2013.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using Dropconnect. In *International Conference on Machine Learning*, pages 1058–1066, 2013.

Colin Wei, Sham Kakade, and Tengyu Ma. The implicit and explicit regularization effects of dropout. In *International Conference on Machine Learning*, pages 10181–10192, 2020.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep ReLU networks. *Machine Learning*, 109(3):467–492, 2020.