On Dropout and Nuclear Norm Regularization

Poorya Mianjy [1] **Raman Arora** [1]

Abstract

We give a formal and complete characterization of the explicit regularizer induced by dropout in deep linear networks with squared loss. We show that (a) the explicit regularizer is composed of an ℓ_2-path regularizer and other terms that are also rescaling invariant, (b) the convex envelope of the induced regularizer is the squared nuclear norm of the network map, and (c) for a sufficiently large dropout rate, we characterize the global optima of the dropout objective. We validate our theoretical findings with empirical results.

1. Introduction

Deep learning is revolutionizing the technological world with recent advances in artificial intelligence. However, a formal understanding of when or why deep learning algorithms succeed has remained elusive. Recently, a series of works focusing on computational learning theoretic aspects of deep learning have implicated inductive biases due to various algorithmic choices to be a crucial potential explanation (Zhang et al., 2016; Gunasekar et al., 2018a; Neyshabur et al., 2014; Martin & Mahoney, 2018; Mianjy et al., 2018). Here, we examine such an implicit bias of dropout in deep linear networks.

Dropout is a popular algorithmic approach that helps training deep neural networks that generalize better (Hinton et al., 2012; Srivastava et al., 2014). Inspired by the reproduction model in the evolution of advanced organisms, dropout training aims at breaking co-adaptation among neurons by dropping them independently and identically according to a Bernoulli random variable.

Here, we restrict ourselves to simpler networks; we consider multi-layered feedforward networks with linear activations (Goodfellow et al., 2016). While the overall function is linear, the representation in factored form makes the optimization landscape non-convex and hence, challenging to analyze. More importantly, we argue that the fact we choose to represent a linear map in a factored form has important implications to the learning problem, akin in many ways to the implicit bias due to stochastic optimization algorithms and various algorithmic heuristics used in deep learning (Gunasekar et al., 2017; Li et al., 2018; Azizan & Hassibi, 2019).

Several recent works have investigated the optimization landscape properties of deep linear networks (Baldi & Hornik, 1989; Saxe et al., 2013; Kawaguchi, 2016; Hardt & Ma, 2016; Laurent & Brecht, 2018), as well as the implicit bias due to first-order optimization algorithms for training such networks (Gunasekar et al., 2018b; Ji & Telgarsky, 2018), and the convergence rates of such algorithms (Bartlett et al., 2018; Arora et al., 2018). The focus here is to have a similar development for dropout when training deep linear networks.

1.1. Notation

For an integer i, $[i]$ represents the set $\{1, \ldots, i\}$, e_i denotes the i-th standard basis, and $1_i \in \mathbb{R}^i$ is the vector of all ones. The set of all k-combinations of a set $\mathcal{S}$ is denoted by $\binom{\mathcal{S}}{k}$. We denote linear operators and vectors by Roman capital and lowercase letters, respectively, e.g. Y and y. Scalar variables are denoted by lower case letters (e.g. y) and sets by script letters, e.g. $\mathcal{Y}$. We denote the ℓ_2 norm of a vector x by $\|\mathrm{x}\|$. For a matrix X, $\|\mathrm{X}\|_F$ denotes the Frobenius norm, $\|\mathrm{X}\|_*$ denotes the nuclear norm, and $\sigma_i(\mathrm{X})$ is the i-th largest singular value of matrix X. For $\mathrm{X} \in \mathbb{R}^{d_2 \times d_1}$ and a positive definite matrix $\mathrm{C} \in \mathbb{R}^{d_1 \times d_1}$, $\|\mathrm{X}\|_{\mathrm{C}}^2 := \mathrm{Tr}\left(\mathrm{XCX}^\top\right)$. The standard inner product between two matrices X, X', is denoted by $\langle \mathrm{X}, \mathrm{X}' \rangle := \mathrm{Tr}\left(\mathrm{X}^\top \mathrm{X}'\right)$. We denote the i-th column and the j-th row of a matrix X with vectors $\mathrm{x}_{:i}$ and $\mathrm{x}_{j:}$, both in column form. The vector of diagonal elements of X is denoted as $\mathrm{diag}(\mathrm{X})$. The diagonal matrix with diagonal entries as the elements of a vector x is denoted as $\mathrm{diag}(\mathrm{x})$. With a slight abuse of notation, we use $\{\mathrm{W}_i\}$ as a shorthand for the tuple $(\mathrm{W}_1, \ldots, \mathrm{W}_{k+1})$.

[1] Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA. Correspondence to: Raman Arora <arora@cs.jhu.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

Table 1. Key terms, corresponding symbols, and descriptions.

Term	**Symbol**	**Description**
architecture	$\{d_i\}$	$(d_0, \ldots, d_{k+1})$
implementation	$\{\mathrm{W}_i\}$	$(\mathrm{W}_0, \ldots, \mathrm{W}_{k+1})$
network map	$\mathrm{W}_{k+1\to 1}$	$\mathrm{W}_{k+1}\mathrm{W}_k \cdots \mathrm{W}_1$
population risk	$L(\{\mathrm{W}_i\})$	$\mathbb{E}_{\mathrm{x},\mathrm{y}}[\\|\mathrm{y} - \mathrm{W}_{k+1\to 1}\mathrm{x}\\|^2]$
dropout objective	$L_\theta(\{\mathrm{W}_i\})$	see Equation (1)
explicit regularizer	$R(\{\mathrm{W}_i\})$	$L_\theta(\{\mathrm{W}_i\}) - L(\{\mathrm{W}_i\})$
induced regularizer	$\Theta(\mathrm{M})$	see Equation (2)

1.2. Problem Setup

We consider the hypotheses class of multilayer feed-forward linear networks with input dimension d_0, output dimension d_{k+1}, k hidden layers of widths $d_1, \ldots, d_k$, and linear transformations $\mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}$, for $i = 1, \ldots, k+1$:

$$\mathcal{L}_{\{d_i\}} = \{g : \mathrm{x} \mapsto \mathrm{W}_{k+1} \cdots \mathrm{W}_1 \mathrm{x},\ \mathrm{W}_i \in \mathbb{R}^{d_i \times d_{i-1}}\}.$$

We refer to the set of $k+1$ integers $\{d_i\}_{i=0}^{k+1}$ specifying the width of each layer as the *architecture* of the function class, the set of the weight parameters $\{\mathrm{W}_i\}_{i=1}^{k+1}$ as an *implementation*, or an element of the function class, and $\mathrm{W}_{k+1\to 1} := \mathrm{W}_{k+1}\mathrm{W}_k \cdots \mathrm{W}_1$ as the *network map*.

The focus here is on *deep regression* with dropout under ℓ_2 loss, which is widely used in computer vision tasks, including human pose estimation (Toshev & Szegedy, 2014), facial landmark detection, and age estimation (Lathuilière et al., 2019). More formally, we study the following learning problem for deep linear networks. Let $\mathcal{X} \subseteq \mathbb{R}^{d_0}$ and $\mathcal{Y} \subseteq \mathbb{R}^{d_{k+1}}$ denote the input feature space and the output label space, respectively. Let $\mathcal{D}$ denote the joint probability distribution on $\mathcal{X} \times \mathcal{Y}$. We assume that $\mathbb{E}[\mathrm{x}\mathrm{x}^\top]$ has full rank. Given a training set $\{\mathrm{x}_i, \mathrm{y}_i\}_{i=1}^n$ drawn i.i.d. from the distribution $\mathcal{D}$, the goal of the learning problem is to minimize the *population risk* under the squared loss $L(\{\mathrm{W}_i\}) := \mathbb{E}_{\mathrm{x},\mathrm{y}}[\|\mathrm{y} - \mathrm{W}_{k+1\to 1}\mathrm{x}\|^2]$. Note that the population risk L depends only on the network map and not the specific implementations of it, i.e. $L(\{\mathrm{W}_i\}) = L(\{\mathrm{W}'_i\})$ for all $\mathrm{W}_{k+1} \cdots \mathrm{W}_1 = \mathrm{W}'_{k+1} \cdots \mathrm{W}'_1$. For that reason, with a slight abuse of notation we write $L(\mathrm{W}_{k+1\to 1}) := L(\{\mathrm{W}_i\})$.

Dropout is an iterative procedure wherein at each iteration each node in the network is dropped independently and identically according to a Bernoulli random variable with parameter θ. Equivalently, we can view dropout, algorithmically, as an instance of stochastic gradient descent for minimizing the following objective over $\{\mathrm{W}_i\}$:

$$L_\theta(\{\mathrm{W}_i\}) := \mathbb{E}_{(\mathrm{x},\mathrm{y},\{\mathrm{b}_i\})}[\|\mathrm{y} - \bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2], \qquad (1)$$

where $\bar{\mathrm{W}}_{i\to j} := \frac{1}{\theta^k}\mathrm{W}_i\mathrm{B}_{i-1}\mathrm{W}_{i-1}\cdots\mathrm{B}_j\mathrm{W}_j$, and $\mathrm{B}_l = \mathrm{diag}\left([\mathrm{b}_l(1), \ldots, \mathrm{b}_l(d_l)]\right)$ represents the dropout pattern in the l^{th} layer with Bernoulli random variables on the diagonal; if $\mathrm{B}_l(i,i) = 0$ then all paths from the input to the output that pass through the i^{th} hidden node in the l^{th} layer are turned "off", i.e., those paths have no contribution in determining the output of the network for that instance of the dropout pattern; we refer to the parameter $1-\theta$ as the *dropout rate*. $\bar{\mathrm{W}}_{i\to j}$ is an unbiased estimator of $\mathrm{W}_{i\to j}$, i.e. $\mathbb{E}_{\{\mathrm{b}_i\}}[\bar{\mathrm{W}}_{i\to j}] = \mathrm{W}_{i\to j}$.

We say that the dropout algorithm *succeeds* in training a network if it returns a map $\mathrm{W}_{k+1\to 1}$ that (approximately) minimizes L_θ. In this paper, the central question under investigation is to understand *which network maps/architectures is a successful dropout training biased towards.*

To answer this question, we begin with the following simple observation that

$$L_\theta(\{\mathrm{W}_i\}) = L(\{\mathrm{W}_i\}) + \mathbb{E}_{(\mathrm{x},\{\mathrm{b}_i\})}\|\mathrm{W}_{k+1\to 1}\mathrm{x} - \bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2$$

In other words, the dropout objective is composed of the population risk $L(\{\mathrm{W}_i\})$ plus an *explicit regularizer* $R(\{\mathrm{W}_i\}) := \mathbb{E}_{(\mathrm{x},\mathrm{y},\{\mathrm{b}_i\})}[\|\mathrm{W}_{k+1\to 1}\mathrm{x} - \bar{\mathrm{W}}_{k+1\to 1}\mathrm{x}\|^2]$ induced by dropout. Denoting the second moment of x by $\mathrm{C} := \mathbb{E}[\mathrm{x}\mathrm{x}^\top]$, we note that $R(\{\mathrm{W}_i\}) = \mathbb{E}_{\{\mathrm{b}_i\}}[\|\mathrm{W}_{k+1\to 1} - \bar{\mathrm{W}}_{k+1\to 1}\|_{\mathrm{C}}^2]$. Since any stochastic network map specified by $\bar{\mathrm{W}}_{k+1\to 1}$ is an unbiased estimator of the network map specified by $\mathrm{W}_{k+1\to 1}$, the explicit regularizer captures the variance of the network implemented by the weights $\{\mathrm{W}_i\}$ under Bernoulli perturbations. By minimizing this variance term, dropout training aims at *breaking co-adaptation between hidden nodes* – it biases towards networks whose random sub-networks yield similar outputs (Srivastava et al., 2014).

If $\{\mathrm{W}_i^*\}$ is an infimum of (1), then it minimizes the explicit regularizer among all implementations of the network map, $\mathrm{M} = \mathrm{W}_{k+1}^* \cdots \mathrm{W}_1^*$, i.e., $R(\{\mathrm{W}_i^*\}) = \inf_{\mathrm{W}_{k+1}\cdots\mathrm{W}_1 = \mathrm{M}} R(\{\mathrm{W}_i\})$. We refer to the infimum of the explicit regularizer over all implementations of a given network map M as the *induced regularizer*:

$$\Theta(\mathrm{M}) := \inf_{\mathrm{W}_{k+1}\cdots\mathrm{W}_1 = \mathrm{M}} R(\{\mathrm{W}_i\}) \qquad (2)$$

The domain of the induced regularizer Θ is the linear maps implementable by the network, i.e., the set $\{\mathrm{M} : \mathrm{rank}\,(\mathrm{M}) \leq \min_i d_i\}$. Since the infimum of the induced regularizer is always attained (see Lemma A.2 in the appendix), we can equivalently study the following problem to understand the solution to Problem 1 in terms of the network map:

$$\min_{\mathrm{M}} L(\mathrm{M}) + \Theta(\mathrm{M}), \quad \mathrm{rank}\,(\mathrm{M}) \leq \min_{i \in [k+1]} d_i. \qquad (3)$$

To characterize which networks are preferable by dropout training, one needs to understand the explicit regularizer

R, understand the induced regularizer Θ, and explore the global minima of Problem 3. In this regard, we make several important contributions summarized as follows.

1. We derive the closed form expression for the explicit regularizer $R(\{\mathrm{W}_i\})$ induced by dropout training in deep linear networks. The explicit regularizer is comprised of the ℓ_2-path regularizer as well as other rescaling invariant sub-regularizers.

2. We show that the convex envelope of the induced regularizer is proportional to the squared nuclear norm of the network map, generalizing a similar result for matrix factorization (Cavazza et al., 2018) and single hidden layer linear networks (Mianjy et al., 2018). Furthermore, we show that the induced regularizer equals its convex envelope if and only if the network is *equalized*, a notion that quantitatively measures *co-adaptation* between hidden nodes (Mianjy et al., 2018).

3. We completely characterize the global minima of the dropout objective L_θ in Problem 1 despite the objective being non-convex, under a simple eigengap condition (see Theorem 2.7). This gap condition depends on the model, the data distribution, the network architecture and the dropout rate, and is always satisfied by deep linear network architectures with one output neuron.

4. We empirically verify our theoretical findings.

The rest of the paper is organized as follows. In Section 2, we present the main results of the paper. In Section 3, we discuss the proof ideas and the key insights. Section 4 details the experimental results and Section 5 concludes with a discussion of future work. We refer the reader to Table 1 for a quick reference to the most useful notation.

2. Main Results

2.1. The explicit regularizer

In this section, we give the closed form expression for the explicit regularizer $R(\{\mathrm{W}_i\})$, and discuss some of its important properties.

Proposition 2.1. *The explicit regularizer is composed of k sub-regularizers: $R(\{\mathrm{W}_i\}) = \sum_{l\in[k]} \lambda^l R_l(\{\mathrm{W}_i\})$, where $\lambda := \frac{1-\theta}{\theta}$. Each of the sub-regularizers has the form:*

$$R_l(\{\mathrm{W}_i\}) = \sum_{\substack{(j_l,\dots,j_1)\\ \in\binom{[k]}{l}}} \sum_{\substack{(i_l,\dots,i_1)\\ \in[d_{j_l}]\times\cdots\times[d_{j_1}]}} \alpha^2_{j_1,i_1} \prod_{p=1\cdots l-1} \beta_p^2 \gamma^2_{j_l,i_l}$$

where $\alpha_{j_1,i_1} := \|\mathrm{W}_{j_1\to 1}(i_1,:)\|_{\mathrm{C}}$, $\beta_p := \mathrm{W}_{j_{p+1}\to j_p+1}(i_{p+1},i_p)$, *and* $\gamma_{j_l,i_l} := \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\|$.

Understanding the regularizer. For simplicity, we assume here the case where the data distribution is whitened, i.e. $\mathrm{C} = \mathrm{I}$. This assumption is by no means restrictive, as we can always redefine $\mathrm{W}_1 \leftarrow \mathrm{W}_1\mathrm{C}^{\frac{1}{2}}$ to absorb the second moment the first layer. Moreover, it is a common practice to whiten the data as a preprocessing step.

The l-th sub-regularizer, i.e. $R_l(\{\mathrm{W}_i\})$, partitions the network graph (see Figure 1) into $l+1$ subgraphs. This partitioning is done via the choice of l *pivot layers*, a set of l distinct hidden layers, indexed by $(j_1,\dots,j_l) \in \binom{[k]}{l}$. The sub-regularizer enumerates over all such combinations of pivot layers, and *pivot nodes* within them indexed by $(i_1,\dots,i_l) \in [d_{j_1}]\times\cdots\times[d_{j_l}]$. For a given set of pivot layers and pivot nodes, the corresponding summand in the sub-regularizer is a product of three types of terms: a "head" term α_{j_1,i_1}, "middle" terms β_p, $p\in[l-1]$, and "tail" terms γ_{j_l,i_l}. It is easy to see that each of the head, middle and tail terms computes a summation over product of the weights along certain walks on the (undirected) graph associated with the network (see Figure 1). For instance, a head term

$$\alpha_{j_1,i_1} = \sum_{i_0\in[d_0]} \sum_{\substack{i'_1,i'_2,\dots,i'_{j_1-1}\\ i''_{j_1-1},\dots,i''_2,i''_1}} \mathrm{W}_1(i'_1,i_0)\mathrm{W}_2(i'_2,i'_1)\cdots$$
$$\mathrm{W}_{j_1}(i_1,i'_{j_l-1})\mathrm{W}_{j_1}(i_1,i''_{j_l-1})\cdots\mathrm{W}_2(i''_2,i''_1)\mathrm{W}_1(i''_1,i_0),$$

is precisely the sum of the product of all weights along all walks from i_0 in the input layer to i_1 in layer j_1 and back to i_0, i.e., walks from $i_0 \xrightarrow{i'_1,i'_2,\dots,i'_{j_1-1}} i_1 \xrightarrow{i''_{j_1-1},\dots,i''_2,i''_1} i_0$. Similarly, middle terms are the sum of the product of the weights along $i_p \xrightarrow{i'_1,i'_2,\dots,i'_{j_1-1}} i_{p+1} \xrightarrow{i''_{j_1-1},\dots,i''_2,i''_1} i_p$.

A few remarks are in order.

Remark 2.2. *For $k=1$, the explicit regularizer reduces to*

$$R(\mathrm{W}_2,\mathrm{W}_1) = \lambda\sum_{i=1}^{d_1} \|\mathrm{W}_1(:,i)\|^2\|\mathrm{W}_2(i,:)\|^2,$$

which recovers the regularizer studied by the previous work of Cavazza et al. (2018) and Mianjy et al. (2018) in the setting of matrix factorization and single hidden layer linear networks, respectively.

Remark 2.3. *All sub-regularizers, and consequently the explicit regularizer itself are* rescaling invariant. *That is, for any given implementation $\{\mathrm{W}_i\}$, and any sequence of scalars $\alpha_1,\dots,\alpha_{k+1}$ such that $\prod_i \alpha_i = 1$, it holds that $R_l(\{\mathrm{W}_i\}) = R_l(\{\alpha_i\mathrm{W}_i\})$. In particular, R_k equals*

$$R_k(\{\mathrm{W}_i\}) = \sum_{i_k,\dots,i_1} \|\mathrm{W}_1(i_1,:)\|^2\mathrm{W}_2(i_2,i_1)^2$$
$$\mathrm{W}_3(i_3,i_2)^2\cdots\mathrm{W}_k(i_k,i_{k-1})^2\|\mathrm{W}_{k+1}(:,i_k)\|^2.$$

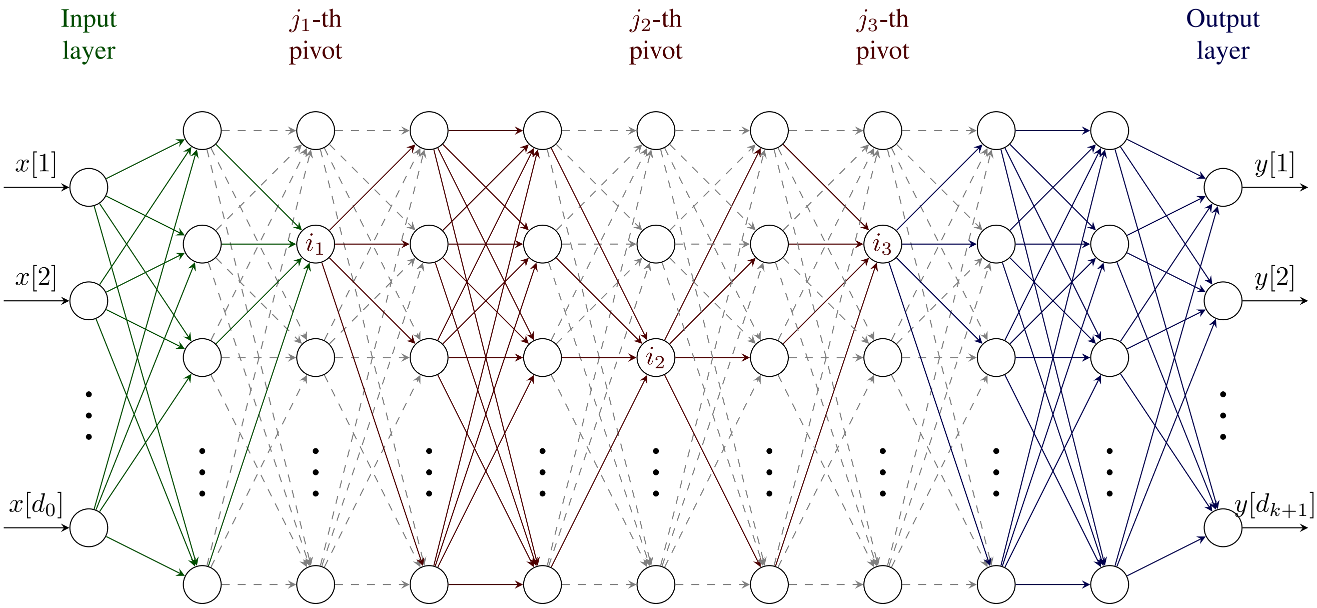

Figure 1. Illustration of the explicit regularizer as given in Proposition 2.1 for $k = 9$, $l = 3$, $(i_1, i_2, i_3) = (2, 3, 2)$ and $(j_1, j_2, j_3) = (2, 5, 7)$. The head term $\alpha^2_{j_1,i_1}$ corresponds to the summation over the product of the weights on any pairs of path from an input node to i_1-th node in the j_1-th hidden layer. Similarly, the tail term $\gamma^2_{j_l,i_l}$ accounts for the product of the weights along any pair of path between the output and the i_l-th node in the j_l-th hidden layer. Each of the middle terms β^2_p, accumulates the product of of the weights along pair of path between i_p-th node in the $(j_p + 1)$-th hidden layer and the i_{p+1}-th node in the j_{p+1}-th hidden layer.

Note that $R_k(\{\mathrm{W}_i\}) = \psi_2^2(\mathrm{W}_{k+1}, \dots, \mathrm{W}_1)$, *the* ℓ_2-path regularizer, *which which has been recently studied in (Neyshabur et al., 2015b) and (Neyshabur et al., 2015a).*

2.2. The induced regularizer

In this section, we study the induced regularizer as given by the optimization problem in Equation (2). We show that the convex envelope of Θ factors into a product of two terms: a term that only depends on the network architecture and the dropout rate, and a term that only depends on the network map. These two factors are captured by the following definitions.

Definition 2.4 (effective regularization parameter)**.** *For given* $\{d_i\}$ *and* λ*, we refer to the following quantity as the effective regularization parameter:*

$$\nu_{\{d_i\}} := \sum_{l \in [k]} \sum_{(j_l, \dots, j_1) \in \binom{[k]}{l}} \frac{\lambda^l}{\prod_{i \in [l]} d_{j_i}}.$$

We drop the subscript $\{d_i\}$ *whenever it is clear from the context.*

The effective regularization parameter naturally arises when we lowerbound the explicit regularizer (see Equation (5)). It is only a function of the network architecture and the dropout rate and does not depend on the weights – it increases with the dropout rate and the depth of the network, but decreases with the width.

Definition 2.5 (equalized network)**.** *A network implemented by* $\{\mathrm{W}_i\}_{i=1}^{k+1}$ *is said to be* equalized *if* $\|\mathrm{W}_{k+1} \cdots \mathrm{W}_1 \mathrm{C}^{\frac{1}{2}}\|_*$ *is equally distributed among all the summands in Proposition 2.1, i.e. for any* $l \in [k]$, $(j_l, \dots, j_1) \in \binom{[k]}{l}$, *and* $(i_l, \dots, i_1) \in [d_{j_l}] \times \dots \times [d_{j_1}]$ *it holds that*

$$|\alpha_{j_1,i_1}\beta_1 \cdots \beta_{l-1}\gamma_{j_l,i_l}| = \frac{\|\mathrm{W}_{k+1} \cdots \mathrm{W}_1 \mathrm{C}^{\frac{1}{2}}\|_*}{\Pi_l d_{j_l}}.$$

We are now ready to state the main result of this section. Recall that the convex envelope of a function is the largest convex under-estimator of that function. We show that irrespective of the architecture, the convex envelope of the induced regularizer is proportional to the squared nuclear norm of the network map times the principal root of the second moment.

Theorem 2.6 (Convex Envelope)**.** *For any architecture* $\{d_i\}$ *and any network map* $\mathrm{M} \in \mathbb{R}^{d_{k+1} \times d_0}$ *implementable by that architecture, it holds that:*

$$\Theta^{**}(\mathrm{M}) = \nu_{\{d_i\}} \|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$$

Furthermore, $\Theta(\mathrm{M}) = \Theta^{**}(\mathrm{M})$ *if and only if the network is equalized.*

This result is particularly interesting because it connects dropout, an algorithmic heuristic to avoid overfitting, to nuclear norm regularization, which is a classical regularization method with strong theoretical foundations. We remark that a result similar to Theorem 2.6 was recently established for matrix factorization (Cavazza et al., 2018).

2.3. Global optimality

Theorem 2.6 provides a sufficient and necessary condition under which the induced regularizer equals its convex envelope. If any network map can be implemented by an equalized network, then $\Theta(\mathrm{M}) = \Theta^{**}(\mathrm{M}) = \nu_{\{d_i\}}\|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$, and the learning problem in Equation (3) is a convex program. In particular, for the case of linear networks with single hidden layer, Mianjy et al. (2018) show that any network map can be implemented by an equalized network, which enables them to characterize the set of global optima under the additional generative assumption $\mathrm{y} = \mathrm{M}\mathrm{x}$. However, it is not clear if the same holds for general deep linear networks since the regularizer here is more complex. Nonetheless, the following result provides a sufficient condition under which global optima of $L_\theta(\{\mathrm{W}_i\})$ are completely characterized.

Theorem 2.7. *Let* $\mathrm{C}_{\mathrm{yx}} := \mathbb{E}[\mathrm{y}\mathrm{x}^\top]$ *and* $\mathrm{C} := \mathbb{E}[\mathrm{x}\mathrm{x}^\top]$, *and denote* $\bar{\mathrm{M}} := \mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}}$. *If* $\sigma_1(\bar{\mathrm{M}}) - \sigma_2(\bar{\mathrm{M}}) \geq \frac{1}{\nu}\sigma_2(\bar{\mathrm{M}})$, *then* M^*, *the global optimum of Problem 3, is given by*

$$\mathrm{W}_{k+1\to 1}^* = \mathcal{S}_{\frac{\nu\sigma_1(\bar{\mathrm{M}})}{1+\nu}}(\bar{\mathrm{M}})\mathrm{C}^{-\frac{1}{2}},$$

where $\mathcal{S}_\alpha(\bar{\mathrm{M}})$ *shrinks the spectrum of matrix* $\bar{\mathrm{M}}$ *by* α *and thresholds it at zero. Furthermore, it is possible to implement* M^* *by an equalized network* $\{\mathrm{W}_i^*\}$ *which is a global optimum of* $L_\theta(\{\mathrm{W}_i\})$.

The gap condition in the theorem above can always be satisfied (e.g. by increasing the dropout rate or the depth, or decreasing the width) as long as there exists a gap between the first and the second singular values of $\bar{\mathrm{M}}$. Moreover, for the special case of deep linear networks with one output neuron (Ji & Telgarsky, 2018; Nacson et al., 2018), this condition is always satisfied since $\bar{\mathrm{M}} \in \mathbb{R}^{1\times d_0}$ and $\sigma_2(\bar{\mathrm{M}}) = 0$.

Corollary 2.8. *Consider the class of deep linear networks with a single output neuron. Let* $\{\mathrm{W}_i^*\}$ *be a minimizer of* L_θ. *For any architecture* $\{d_i\}$ *and any network map* $\mathrm{W}_{k+1\to1}$, *it holds that: (1)* $\Theta(\mathrm{W}_{k+1\to1}) = \nu\|\mathrm{W}_{k+1\to1}\|_{\mathrm{C}}^2$, *(2)* $\mathrm{W}_{k+1\to1}^* = \frac{1}{1+\nu}\mathrm{C}_{\mathrm{yx}}$, *(3) the network specified by* $\{\mathrm{W}_i^*\}$ *is equalized.*

We conclude this section with a remark. We know from the early work of (Srivastava et al., 2014) that feature dropout in linear regression is closely related to ridge regression. Corollary 2.8 generalizes the results of (Srivastava et al., 2014) to deep linear networks, and establishes a similar connection between dropout training and ridge regression.

3. Proof Ideas

Here, we sketch proofs of the main results from Section 2; complete proofs are deferred to the supplementary.

Sketch of the Proof of Theorem 2.6 The key steps are:

1. First, in Lemma 3.1, we show that for any set of weights $\{\mathrm{W}_i\}$, it holds that $R(\{\mathrm{W}_i\}) \geq \nu_{\{d_i\}}\|\mathrm{W}_{k+1\to1}\mathrm{C}^{\frac{1}{2}}\|_*^2$. In particular, this implies that $\Theta(\mathrm{M}) \geq \nu_{\{d_i\}}\|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$ holds for any M.
2. Next, in Lemma 3.2, we show that $\Theta^{**}(\mathrm{M}) \leq \nu_{\{d_i\}}\|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$ holds for all M.
3. The claim is implied by Lemmas 3.1 and 3.2, and the fact that $\|\cdot\|_*^2$ is a convex function.

Lemma 3.1. *Let* $\{\mathrm{W}_i\}$ *be an arbitrary set of weights. The explicit regularizer* $R(\{\mathrm{W}_i\})$ *satisfies*

$$R(\{\mathrm{W}_i\}) \geq \nu_{\{d_i\}}\|\mathrm{W}_{k+1}\mathrm{W}_k\cdots\mathrm{W}_1\mathrm{C}^{\frac{1}{2}}\|_*^2,$$

and the equality holds if and only if the network is equalized.

We sketch the proof for isotropic distributions ($\mathrm{C} = \mathrm{I}$), and emphasize the role of equalization and effective regularization parameter.

Sketch of the Proof of Lemma 3.1 We show that each term in the explicit regularizer is lower bounded by some multiple of the square of the nuclear norm of the linear map implemented by the network. We begin by lower bounding a particular summand in the expansion of $R_l(\{\mathrm{W}_i\})$ given in Proposition 2.1 with indices $j_1, \ldots, j_l$:

$$R_{\{j_l,\ldots,j_1\}}(\{\mathrm{W}_i\}) = \sum_{i_l,\ldots,i_1} \alpha_{j_1,i_1}^2 \prod_{p=1\cdots l-1} \beta_p^2\gamma_{j_l,i_l}^2$$
$$\geq \frac{1}{\Pi_l d_{j_l}}\left(\sum_{i_l,\ldots,i_1} |\alpha_{j_1,i_1}\beta_1\cdots\beta_{l-1}\gamma_{j_l,i_l}|\right)^2,$$

where the inequality follows due to Cauchy-Schwartz inequality and holds with equality if and only if all the summands in $\phi := \sum_{i_l,\ldots,i_1}|\alpha_{j_1,i_1}\beta_1\cdots\beta_{l-1}\gamma_{j_l,i_l}|$ are equal for all $i_l,\ldots,i_1 \in [d_{j_l}]\times\cdots\times[d_{j_1}]$. At the same time, we have that

$$\phi = \sum_{i_l,\ldots,i_1} \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\||\beta_1\cdots\beta_{l-1}|\|\mathrm{W}_{j_1\to1}(i_1,:)\|$$
$$\overset{(a)}{=} \sum_{i_l,\ldots,i_1} \|\mathrm{W}_{k+1\to j_l+1}(:,i_l)\beta_1\cdots\beta_{l-1}\mathrm{W}_{j_1\to1}(i_1,:)^\top\|_*$$
$$\geq \|\sum_{i_l,\ldots,i_1} \mathrm{W}_{k+1\to j_l+1}(:,i_l)\beta_1\cdots\beta_{l-1}\mathrm{W}_{j_1\to1}(i_1,:)^\top\|_*$$
$$\overset{(b)}{=} \|\mathrm{W}_{k+1}\cdots\mathrm{W}_1\|_*$$

where (a) follows since the outer product inside the nuclear norm has rank equal to 1, the inequality is due to the triangle inequality for matrix norms, and (b) follows trivially. In fact, the inequality above holds if each of the summands in (a) are equal to

$$|\alpha_{j_1,i_1}\beta_1\cdots\beta_{l-1}\gamma_{j_l,i_l}| = \frac{\|\mathrm{W}_{k+1}\cdots\mathrm{W}_1\|_*}{\Pi_l d_{j_l}} \tag{4}$$

for all $i_l,\ldots,i_1 \in [d_{j_l}]\times\cdots\times[d_{j_1}]$. Each of the sub-regularizers can be lowerbounded by noting that $R_l(\{\mathrm{W}_i\}) = \sum_{j_l,\ldots,j_1} R_{\{j_l,\ldots,j_1\}}(\{\mathrm{W}_i\})$.

Lemma 3.1 is central to our analysis for two reasons. First, it gives a sufficient and necessary condition for the induced regularizer to equal the square of the nuclear norm of the network map. This also motivates the concept of equalized networks in Definition 2.5. We note that for the special case of single hidden layer linear networks, i.e., for k=1, this lower bound can always be achieved (Mianjy et al., 2018); it remains to be seen whether the lower bound can be achieved for deeper networks. Second, summing over $\{j_l,\ldots,j_1\}\in\binom{[k]}{l}$, we conclude that

$$R_l(\{\mathrm{W}_i\}) \geq \sum_{j_l,\ldots,j_1} \frac{\|\mathrm{W}_{k+1\to 1}\|_*^2}{\prod_l d_{j_l}} =: LB_l(\{W_i\}). \tag{5}$$

The right hand side above is the lowerbound for l-th subregularizer, denoted by LB_l. Summing over $l\in[k]$, we get the following lowerbound on the explicit regularizer

$$R(\{\mathrm{W}_i\}) \;\geq\; \|\mathrm{W}_{k+1\to 1}\|_*^2 \underbrace{\sum_l \sum_{j_l,\ldots,j_1} \frac{\lambda^l}{\prod_l d_{j_l}}}_{\nu_{\{d_i\}}} \tag{6}$$

which motivates the notion of *effective regularization parameter* in Definition 2.4. As an immediate corollary of Lemma 3.1, it holds that for any matrix M we have that $\Theta(\mathrm{M}) \geq \nu_{\{d_i\}}\|\mathrm{M}\|_*^2$. We now focus on the biconjugate of the induced regularizer, and show that it is upper bounded by the same function, i.e. the effective regularization parameter times the square of the nuclear norm of the network map.

Lemma 3.2. *For any architecture $\{d_i\}$ and any network map* M*, it holds that* $\Theta^{**}(\mathrm{M}) \leq \nu_{\{d_i\}}\|\mathrm{M}\mathrm{C}^{\frac{1}{2}}\|_*^2$.

To convey the main ideas of the proof, here we include a sketch for the simple case of $k=2$, $d_1=d_2=d$ under the isotropic assumption ($\mathrm{C}=\mathrm{I}$); for the general case, please refer to the appendix.

Sketch of the proof of Lemma 3.2 First, the induced regularizer is non-negative, so the domain of Θ^* is $\mathbb{R}^{d_{k+1}\times d_0}$. The Fenchel dual of the induced regularizer $\Theta(\cdot)$ is given by:

$$\begin{aligned}\Theta^*(\mathrm{M}) &= \max_{\mathrm{P}} \langle \mathrm{M},\mathrm{P}\rangle - \Theta(\mathrm{P})\\ &= \max_{\mathrm{P}} \langle \mathrm{M},\mathrm{P}\rangle - \min_{\substack{\mathrm{W}_3,\mathrm{W}_2,\mathrm{W}_1\\ \mathrm{W}_{3\to1}=\mathrm{P}}} R(\mathrm{W}_3,\mathrm{W}_2,\mathrm{W}_1)\\ &= \max_{\mathrm{W}_3,\mathrm{W}_2,\mathrm{W}_1} \langle \mathrm{M},\mathrm{W}_{3\to1}\rangle - R(\mathrm{W}_3,\mathrm{W}_2,\mathrm{W}_1), \quad (7)\end{aligned}$$

Denote the objective in (7) by $\Phi(\mathrm{W}_3,\mathrm{W}_2,\mathrm{W}_1) := \langle \mathrm{M},\mathrm{W}_{3\to1}\rangle - R(\mathrm{W}_3,\mathrm{W}_2,\mathrm{W}_1)$. Let $(\mathrm{u}_1,\mathrm{v}_1)$ be the top singular vectors of M. For any $\alpha\in\mathbb{R}$, consider the following assignments to the weights: $\mathrm{W}_1^\alpha = \alpha\mathrm{u}_1 1_d^\top$, $\mathrm{W}_2^\alpha = 1_d 1_d^\top$ and $\mathrm{W}_3^\alpha = 1_d\mathrm{v}_1^\top$. Note that

$$\max_{\mathrm{W}_3,\mathrm{W}_2,\mathrm{W}_1} \Phi(\mathrm{W}_3,\mathrm{W}_2,\mathrm{W}_1) \geq \max_\alpha \Phi(\mathrm{W}_3^\alpha,\mathrm{W}_2^\alpha,\mathrm{W}_1^\alpha).$$

We can express the objective on the right hand side merely in terms of α, d and $\|\mathrm{M}\|_2$ as follows:

$$\begin{aligned}&R(\mathrm{W}_3^\alpha,\mathrm{W}_2^\alpha,\mathrm{W}_1^\alpha) = \lambda\sum_{i=1}^d \|\mathrm{W}_1^\alpha(i,:)\|^2\|\mathrm{W}_{3\to2}^\alpha(:,i)\|^2\\ &+\lambda\sum_{i=1}^d \|\mathrm{W}_{2\to1}^\alpha(i,:)\|^2\|\mathrm{W}_3^\alpha(:,i)\|^2\\ &+\lambda^2\sum_{i,j=1}^d \|\mathrm{W}_1^\alpha(i,:)\|^2\mathrm{W}_2^\alpha(j,i)^2\|\mathrm{W}_3^\alpha(:,j)\|^2\\ &= 2\lambda\alpha^2 d^3 + \lambda^2\alpha^2 d^2.\end{aligned}$$

Similarly, the inner product $\langle \mathrm{M},\mathrm{W}_{3\to1}^\alpha\rangle$ reduces to

$$\begin{aligned}\langle \mathrm{M},\mathrm{W}_{3\to1}^\alpha\rangle &= \sum_{i,j=1}^d \langle \mathrm{M},\mathrm{W}_3^\alpha(:,j)\mathrm{W}_2^\alpha(j,i)\mathrm{W}_1^\alpha(i,:)^\top\rangle\\ &= \sum_{i,j=1}^d \alpha\mathrm{u}_1^\top\mathrm{M}\mathrm{v}_1 = \alpha d^2\|\mathrm{M}\|_2.\end{aligned}$$

Plugging back the above equalities in Φ we get:

$$\Phi(\mathrm{W}_3^\alpha,\mathrm{W}_2^\alpha,\mathrm{W}_1^\alpha) = \alpha d^2\|\mathrm{M}\|_2 - 2\lambda\alpha^2 d^3 - \lambda^2\alpha^2 d^2.$$

Maximizing the right hand side above with respect to α

$$\Theta^*(\mathrm{M}) \geq \max_\alpha \Phi(\mathrm{W}_3^\alpha,\mathrm{W}_2^\alpha,\mathrm{W}_1^\alpha) = \frac{d^2}{4(2\lambda d+\lambda^2)}\|\mathrm{M}\|_2^2.$$

Since Fenchel dual is order reversing, we get

$$\Theta^{**}(\mathrm{M}) \leq \frac{2\lambda d+\lambda^2}{d^2}\|\mathrm{M}\|_*^2 = \nu_{\{d,d\}}\|\mathrm{M}\|_*^2, \tag{8}$$

where we used the basic duality between the spectral norm and the nuclear norm. Lemma 3.1 implies that the biconjugate $\Theta^{**}(\mathrm{M})$ is lower bounded by $\nu_{\{d,d\}}\|\mathrm{M}\|_*^2$, which is a convex function. On the other hand, inequality (8) shows that the biconjugate is upper bounded $\nu_{\{d,d\}}\|\mathrm{M}\|_*^2$. Since square of the nuclear norm is a convex function, and that $\Theta^{**}(\cdot)$ is the largest convex function that lower bounds $\Theta(\cdot)$, we conclude that $\Theta^{**}(\mathrm{M}) = \nu_{\{d,d\}}\|\mathrm{M}\|_*^2$.

Sketch of the proof of Theorem 2.7 In light of Theorem 2.6, if the optimal network map $\mathrm{W}_{k+1\to1}^*$, i.e. the optimum of the problem in Equation 3 can be implemented by an equalized network, then $\Theta(\mathrm{W}^*_{k+1\to 1}) = \Theta^{**}(\mathrm{W}^*_{k+1\to 1}) = \nu_{\{d_i\}}\|\mathrm{W}^*_{k+1\to 1}\mathrm{C}^{\frac{1}{2}}\|^2_*$. Thus, the learning problem essentially boils down to the following convex program:

$$\min_{\mathrm{W}} \mathbb{E}_{\mathrm{x},\mathrm{y}}[\|\mathrm{y} - \mathrm{W}\mathrm{x}\|^2] + \nu_{\{d_i\}}\|\mathrm{W}\mathrm{C}^{\frac{1}{2}}\|^2_*. \qquad (9)$$

Following the previous work of (Cavazza et al., 2018; Mianjy et al., 2018), we show that the solution to problem (9) can be given as $\mathrm{W}^* = \mathcal{S}_{\alpha_\rho}(\mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}})\mathrm{C}^{-\frac{1}{2}}$, where $\alpha_\rho := \frac{\nu\sum_{i=1}^{\rho}\sigma_i(\mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}})}{1+\rho\nu}$, ρ is the rank of W^*, and $\mathcal{S}_{\alpha_\rho}(\mathrm{M})$ shrinks the spectrum of the input matrix M by α_ρ and thresholds them at zero. However, as mentioned above, it is not clear if any network map can be implemented by an equalized network. Nonetheless, it is easy to see that the equalization property is satisfied for rank-1 network maps.

Proposition 3.3. *Let $\{d_i\}_{i=0}^{k+1}$ be an architecture and $\mathrm{M} \in \mathbb{R}^{d_{k+1}\times d_0}$ be a rank-1 network map. Then, there exists a set of weights $\{\mathrm{W}_i\}_{i=1}^{k+1}$ that implements M, and is equalized.*

For example, for deep networks with single output neuron, the weights $\mathrm{W}_1 = \frac{1_{d_1}\mathrm{w}^\top}{\sqrt{d_1}}$ and $\mathrm{W}_i = \frac{1_{d_i}1_{d_{i-1}}^\top}{\sqrt{d_i d_{i-1}}}$ for $i \neq 1$ implements the network map $\mathrm{w}^\top$, and are equalized.

Denote $\bar{\mathrm{M}} := \mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}}$. Equipped with Proposition 3.3, the key here is to ensure that $\mathcal{S}_\alpha(\bar{\mathrm{M}})$ has rank equal to one. In this case, W^* will also have rank at most one and can be implemented by a network that is equalized. To that end, it suffices to have $\alpha \geq \sigma_2(\bar{\mathrm{M}})$, which implies

$$\frac{\nu\sigma_1(\bar{\mathrm{M}})}{1+\nu} \geq \sigma_2(\bar{\mathrm{M}}) \implies \sigma_1(\bar{\mathrm{M}}) - \sigma_2(\bar{\mathrm{M}}) \geq \frac{\sigma_2(\bar{\mathrm{M}})}{\nu}$$

which gives the sufficient condition in Theorem 2.7.

4. Experimental Results

Dropout is widely used for training modern deep learning architectures resulting in the state-of-the-art performance in numerous machine learning tasks (Srivastava et al., 2014; Krizhevsky et al., 2012; Szegedy et al., 2015; Toshev & Szegedy, 2014). The purpose of this section is not to make a case for (or against) dropout when training deep networks, but rather verify empirically the theoretical results from the previous section.[1]

For simplicity, the training data $\{\mathrm{x}_i\}$ is sampled from a standard Gaussian distribution which in particular ensures that $\mathrm{C} = \mathrm{I}$. The labels $\{\mathrm{y}_i\}$ are generated as $\mathrm{y}_i \leftarrow \mathrm{N}\mathrm{x}_i$, where $\mathrm{N} \in \mathbb{R}^{d_{k+1}\times d_0}$. N is composed of $\mathrm{U}\mathrm{V}^\top + \texttt{noise}$, where $\mathrm{U} \in \mathbb{R}^{d_{k+1}\times r}$, $\mathrm{V} \in \mathbb{R}^{d_0\times r}$ are sampled from a standard Gaussian and the entries of noise are sampled independently from a Gaussian distribution with small standard deviation. At each step of the dropout training, we use a minibatch of size 1000 to train the network. The learning rate is tuned over the set $\{1, 0.1, 0.01\}$. All experiments are repeated 50 times, the curves correspond to the average of the runs, and the grey region shows the standard deviation.

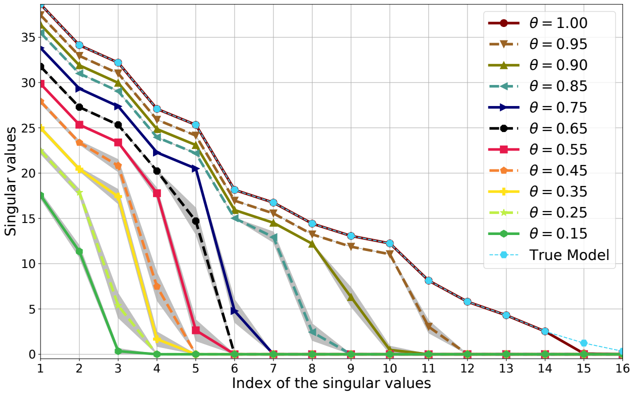

Figure 2. Distribution of the singular values of the trained network for different values of the dropout rate $1-\theta$. It can be seen that the dropout training performs a more sophisticated form of shrinkage and thresholding on the spectrum of the model matrix M.

The experiments are organized as follows. First, since the convex envelope of the induced regularizer equals the squared nuclear norm of the network map (Theorem 2.6), it is natural to expect that dropout training performs a shrinkage-thresholding on the spectrum of $\mathrm{C}_{\mathrm{yx}}\mathrm{C}^{-\frac{1}{2}} = \mathrm{M}$ (see Lemma A.4 in the appendix). We experimentally verify this in Section 4.1. Second, in Section 4.2, we focus on the equalization property. We attest Theorem 2.7 by showing that dropout training equalizes deep networks with one output neuron.

4.1. Spectral shrinkage and rank control

Note that the induced regularizer $\Theta(\mathrm{M})$ is a *spectral function* (see Lemma A.2 in the appendix). On the other hand, by Theorem 2.6, $\Theta^{**}(\mathrm{M}) = \nu_{\{d_i\}}\|\mathrm{M}\|^2_*$. Therefore, if dropout training succeeds in finding an (approximate) minimizer of L_θ, it minimizes an upperbound on the squared of the nuclear norm of the network map. Hence, it is natural to expect that the dropout training performs a shrinkage-thresholding on the spectrum of the model, much like nuclear norm regularization. Figure 2 confirms this intuition. Here, we plot the singular value distribution of the final network map trained by dropout, for different values of the dropout rate.

As can be seen in the figure, dropout training indeed shrinks the spectrum of the model and thresholds it at zero. However, unlike the nuclear norm regularization, the shrinkage is not uniform across the singular values that are not

[1] The code for the experiments can be found at: https://github.com/r3831/dln_dropout

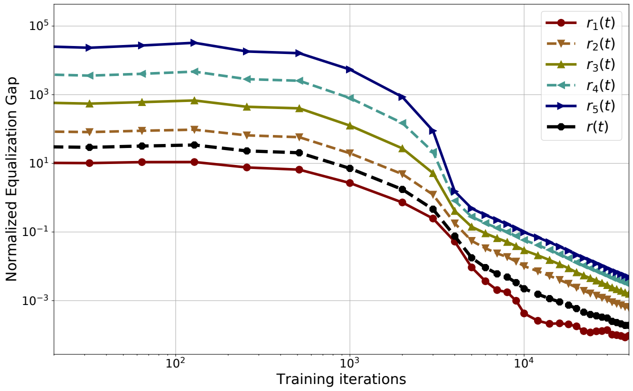

Figure 3. The normalized equalization gap $r_\ell^{(t)}$, which captures the gap between the sub-regularizers and their respective lower bounds, is plotted as a function of the number of iterations. Dropout converges to the set of equalized networks.

thresholded. Moreover, note that the shrinkage parameter in Theorem 2.7 is governed by the effective regularization parameter $\nu_{\{d_i\}}$, which strictly increases with the dropout rate. This suggests that as we increase the dropout rate (decrease θ), the spectrum should be shrunk more severely, and the resulting network map should have a smaller rank. This is indeed the case as can be seen in Figure 2.

4.2. Convergence to equalized networks

One of the key concepts behind our analysis is the notion of equalized networks. In particular, in Lemma 3.1 we see that if a network map can be implemented by an equalized network, then there is no gap between the induced regularizer and its convex envelope. It is natural to ask if dropout training indeed finds such equalized networks. As we will discuss, Figure 3 answers this question affirmatively.

Recall that a network is equalized if and only if each and every sub-regularizer achieves its respective lowerbound in Equation 5, i.e. $R_l(\{\mathrm{W}_i\}) = LB_l(\{\mathrm{W}_i\})$ for all $l \in [k]$. Figure 3 illustrates that dropout training consistently decreases the gap between the sub-regularizers and their respective lowerbounds. Here, the network has one output neuron, five hidden layers each of width 5, and input dimensionality of $d_0 = 5$. In Figure 3 we plot the *normalized equalization gap* $r_\ell^{(t)} := \frac{R_\ell(\{\mathrm{W}_i^{(t)}\})}{LB_\ell(\{\mathrm{W}_i^{(t)}\})} - 1$ of the network under dropout training as a function of the iteration number. Similarly, we define the normalized equalization gap for the explicit regularizer $r^{(t)} = \frac{R(\{\mathrm{W}_i\})}{\Theta^{**}(\mathrm{W}_{k+1\to 1})} - 1$. The network quickly becomes (approximately) equalized, and thereafter the trajectory of dropout training stays close to the equalized networks. We believe that this observation can be helpful in analyzing the dynamics of dropout training, which we leave for future work.

5. Discussion

Motivated by empirical success of dropout (Srivastava et al., 2014; Krizhevsky et al., 2012), there has been several studies in recent years to understand its theoretical foundations (Baldi & Sadowski, 2013; Wager et al., 2013; 2014; Van Erven et al., 2014; Helmbold & Long, 2015; Gal & Ghahramani, 2016; Gao & Zhou, 2016; Helmbold & Long, 2017; Mou et al., 2018; Bank & Giryes, 2018).

Previous work of Zhai & Zhang (2015); He et al. (2016); Cavazza et al. (2018) and Mianjy et al. (2018) study dropout training with ℓ_2-loss in matrix factorization and shallow linear networks, respectively. The work that is most relevant to us is that of Cavazza et al. (2018); Mianjy et al. (2018), whose results are extended to the case of deep linear networks in this paper. In particular, we derive the *explicit regularizer* induced by dropout, which happens to be composed of the ℓ_2-path regularizer and other rescaling invariant regularizers. Furthermore, we show that the convex envelope of the induced regularizer factors into an *effective regularization parameter* and the square of the nuclear norm of network map multiplied with the principal root of the second moment of the input distribution. We further highlight *equalization* as a key network property under which the induced regularizer equals its convex envelope. We specify a subclass of problems satisfying the equalization property, for which we completely characterize the optimal networks that dropout training is biased towards.

Our work suggests several interesting directions for future research. First, given the connections that we establish with the nuclear norm and the ℓ_2-path regularization, it is natural to ask what role does the dropout regularizer play in generalization. Second, how does the dropout regularizer change for neural networks with non-linear activation functions. Finally, it is important to understand dropout in networks trained with other loss functions, especially those that are popular for various classification tasks.

Acknowledgements

This research was supported in part by NSF BIGDATA grant IIS-1546482.

References

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

Azizan, N. and Hassibi, B. Stochastic gradient/mirror descent: Minimax optimality and implicit regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

Baldi, P. and Sadowski, P. J. Understanding dropout. In *Adv. Neural Information Processing Systems*, 2013.

Bank, D. and Giryes, R. On the relationship between dropout and equiangular tight frames. *arXiv preprint arXiv:1810.06049*, 2018.

Bartlett, P., Helmbold, D., and Long, P. Gradient descent with identity initialization efficiently learns positive definite linear transformations. In *ICML*, 2018.

Cavazza, J., Haeffele, B. D., Lane, C., Morerio, P., Murino, V., and Vidal, R. Dropout as a low-rank regularizer for matrix factorization. *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Int. Conf. Machine Learning (ICML)*, 2016.

Gao, W. and Zhou, Z.-H. Dropout rademacher complexity of deep neural networks. *Science China Information Sciences*, 59(7):072104, 2016.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 6151–6159, 2017.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018a.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. *arXiv preprint arXiv:1806.00468*, 2018b.

Hardt, M. and Ma, T. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

He, Z., Liu, J., Liu, C., Wang, Y., Yin, A., and Huang, Y. Dropout non-negative matrix factorization for independent feature learning. In *Int. Conf. on Computer Proc. of Oriental Languages*. Springer, 2016.

Helmbold, D. P. and Long, P. M. On the inductive bias of dropout. *Journal of Machine Learning Research (JMLR)*, 16:3403–3454, 2015.

Helmbold, D. P. and Long, P. M. Surprising properties of dropout in deep networks. *The Journal of Machine Learning Research*, 18(1):7284–7311, 2017.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

Kawaguchi, K. Deep learning without poor local minima. In *Adv in Neural Information Proc. Systems*, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Lathuilière, S., Mesejo, P., Alameda-Pineda, X., and Horaud, R. A comprehensive analysis of deep regression. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Laurent, T. and Brecht, J. Deep linear networks with arbitrary loss: All local minima are global. In *International Conference on Machine Learning*, pp. 2908–2913, 2018.

Li, Y., Ma, T., and Zhang, H. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pp. 2–47, 2018.

Martin, C. H. and Mahoney, M. W. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *arXiv preprint arXiv:1810.01075*, 2018.

Mianjy, P., Arora, R., and Vidal, R. On the implicit bias of dropout. In *International Conference on Machine Learning*, pp. 3537–3545, 2018.

Mou, W., Zhou, Y., Gao, J., and Wang, L. Dropout training, data-dependent regularization, and generalization bounds. In *International Conference on Machine Learning*, pp. 3642–3650, 2018.

Nacson, M. S., Lee, J., Gunasekar, S., Srebro, N., and Soudry, D. Convergence of gradient descent on separable data. *arXiv preprint arXiv:1803.01905*, 2018.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Neyshabur, B., Salakhutdinov, R. R., and Srebro, N. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2015a.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pp. 1376–1401, 2015b.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1), 2014.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Toshev, A. and Szegedy, C. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653–1660, 2014.

Van Erven, T., Kotłowski, W., and Warmuth, M. K. Follow the leader with dropout perturbations. In *Conference on Learning Theory*, pp. 949–974, 2014.

Wager, S., Wang, S., and Liang, P. S. Dropout training as adaptive regularization. In *Adv. Neural Information Processing Systems*, 2013.

Wager, S., Fithian, W., Wang, S., and Liang, P. S. Altitude training: Strong bounds for single-layer dropout. In *Adv. Neural Information Processing Systems*, 2014.

Zhai, S. and Zhang, Z. Dropout training of matrix factorization and autoencoder for link prediction in sparse graphs. In *ICDM*, pp. 451–459, 2015.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.