Dropout: Explicit Forms and Capacity Control

Raman Arora [1] **Peter Bartlett** [2] **Poorya Mianjy** [1] **Nathan Srebro** [3]

Abstract

We investigate the capacity control provided by dropout in various machine learning problems. First, we study dropout for matrix completion, where it induces a distribution-dependent regularizer that equals the weighted trace-norm of the product of the factors. In deep learning, we show that the distribution-dependent regularizer due to dropout directly controls the Rademacher complexity of the underlying class of deep neural networks. These developments enable us to give concrete generalization error bounds for the dropout algorithm in both matrix completion as well as training deep neural networks.

1. Introduction

Dropout is a popular algorithmic regularization technique for training deep neural networks that aims at "breaking co-adaptation" among neurons by randomly dropping them at training time (Hinton et al., 2012). Dropout has been shown effective across a wide range of machine learning tasks, from classification (Srivastava et al., 2014; Szegedy et al., 2015) to regression (Toshev & Szegedy, 2014). Notably, dropout is considered an essential component in the design of AlexNet (Krizhevsky et al., 2012), which won the ImageNet challenge in 2012 with a significant margin.

Dropout regularizes the empirical risk by randomly perturbing the model parameters during training. A natural first step toward understanding generalization due to dropout, therefore, is to instantiate the explicit form of the regularizer due to dropout. In linear regression, with dropout applied to the input layer (i.e., on the input features), the explicit regularizer was shown to be akin to a data-dependent ridge penalty (Srivastava et al., 2014; Wager et al., 2013; Baldi & Sadowski, 2013; Wang & Manning, 2013). In factored models, dropout yields more exotic forms of regularization. For instance, dropout induces regularizer that behaves similar to nuclear norm regularization in matrix factorization (Cavazza et al., 2018; Mianjy et al., 2018), in two layer linear networks (Mianjy et al., 2018), and in deep linear networks (Mianjy & Arora, 2019). However, none of the works above discuss how the induced regularizer provides capacity control, or help us establish generalization bounds for dropout.

In this paper, we give *explicit forms* of the regularizers induced by dropout for the matrix sensing problem and two-layer neural networks with ReLU activations. Further, we establish *capacity control* due to dropout and give precise generalization bounds. Our key contributions are as follows.

1. Our generalization bounds are solely in terms of the value of the explicit regularizer due to dropout. This is a significant departure from most of the prior work wherein dropout is analyzed in conjunction with additional norm-based capacity control, e.g., max-norm (Wan et al., 2013; Gao & Zhou, 2016), or ℓ_p norm on the weights of the model (Zhai & Wang, 2018).

2. Our generalization bounds are data-dependent. We identify a simple distributional property (a notion we refer to as *retentivity*) that yields tight generalization bounds as evidenced by matching lower and upper bounds. We believe that this property may be useful more generally; see Zhang et al. (2021) for another application.

3. Our results emphasize the role of parametrization, i.e., the choice of model architecture. We find that dropout does not yield useful capacity control when training a two-layer linear networks (unless we further assume that the covariance matrix of input features satisfies certain isotropic assumption). On the other hand, dropout for training a network with convolutional topology or a non-linearity imparts useful inductive bias (see Section 4 for more details).

4. We provide extensive numerical evaluations for validating our theory including verifying that the proposed theoretical bound on the Rademacher complexity is predictive of the observed generalization gap as well as highlighting how dropout breaks "co-adaptation", a notion that was the main motivation behind the invention of dropout (Hinton et al., 2012).

[1] Johns Hopkins University. [2] University of California, Berkeley. [3] TTI Chicago.. Correspondence to: Raman Arora <arora@cs.jhu.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

The rest of the paper is organized as follows.

1. In Section 2, we study dropout for matrix completion, wherein, the matrix factors are dropped randomly during training. We show that this algorithmic procedure induces a data-dependent regularizer that in expectation behaves similar to the weighted trace-norm which has been shown to yield strong generalization guarantees for matrix completion (Foygel et al., 2011).

2. In Section 3, we study dropout in two-layer ReLU networks. We show that the regularizer induced by dropout is a data-dependent measure that in expectation behaves as ℓ_2-path norm (Neyshabur et al., 2015a), and establish distribution-dependent generalization bounds.

3. In Section 5, we present empirical evaluations that confirm our theoretical findings for matrix completion and deep regression on real world datasets including the MovieLens data, as well as the MNIST and Fashion MNIST datasets.

1.1. Related Work

Dropout was first introduced by Hinton et al. (2012) as an effective heuristic for algorithmic regularization, yielding lower test errors on the MNIST and TIMIT datasets. In a subsequent work, Srivastava et al. (2014) reported similar improvements over several tasks in computer vision (on CIFAR-10/100 and ImageNet datasets), speech recognition, text classification and genetics.

Thenceforth, dropout has been widely used in training state-of-the-art systems for several tasks including large-scale visual recognition (Szegedy et al., 2015), large vocabulary continuous speech recognition (Dahl et al., 2013), image question answering (Yang et al., 2016), handwriting recognition (Pham et al., 2014), sentiment prediction and question classification (Kalchbrenner et al., 2014), dependency parsing (Chen & Manning, 2014), and brain tumor segmentation (Havaei et al., 2017).

Following the empirical success of dropout, there have been several studies in recent years aimed at establishing theoretical underpinnings of why and how dropout helps with generalization. Early work of Baldi & Sadowski (2013) showed that for a single linear unit (and a single sigmoid unit, approximately), dropout amounts to weight decay regularization on the weights. A similar result was shown by McAllester (2013) in a PAC-Bayes setting. For generalized linear models, Wager et al. (2013) established that dropout performs an adaptive regularization which is equivalent to a data-dependent scaling of the weight decay penalty. In their follow-up work, Wager et al. (2014) show that for linear classification, under a generative assumption on the data, dropout improves the convergence rate of the generalization error. Finally, Mianjy & Arora (2020) studied dropout in over-parameterized two-layer networks with ReLU activation and gave precise generalization error rates under a data separability assumption. In contrast, this paper focuses on predictors represented in a factored form and give generalization bounds for matrix learning and two layer ReLU networks and does not require over-parameterization or data separability.

In a related line of work, Helmbold & Long (2015) study the structural properties of the dropout regularizer in the context of linear classification. They characterize the landscape of the dropout criterion in terms of unique minimizers and establish non-monotonic and non-convex nature of the regularizer. In follow up work, Helmbold & Long (2017) extend their analysis to dropout in deep ReLU networks and surprisingly find that the nature of regularizer is different from that in linear classification. In particular, they show that unlike weight decay, dropout regularizer in deep networks can grow exponentially with depth and remains invariant to rescaling of inputs, outputs, and network weights. We confirm some of these findings in our theoretical analysis. However, counter to the claims of Helmbold & Long (2017), we argue that dropout does indeed prevent co-adaptation.

Using an approach closely related to ours, several works bound the Rademacher complexity of deep neural networks trained using dropout. In particular, Gao & Zhou (2016), (Wan et al., 2013) and (Zhai & Wang, 2018), all show that Rademacher complexity of the target class decreases, assuming additional norm bounds on the weight vectors. In a recent work, Wei et al. (2020) disentangle the explicit and implicit regularization effects of dropout; i.e. the regularization due to the expected bias that is induced by dropout, versus the regularization induced by the noise due to the randomness in dropout. They propose an approximation of the explicit regularizer for deep neural networks and show it to be effective in practice. Their generalization bounds, however, are limited to linear models and similar to other works we discuss above, require weights to be norm bounded. In this paper, we argue, formally, that dropout training alone does not directly control the norms of the weight vectors. Therefore, we *seek to understand if the expected explicit regularizer alone is sufficient for controlling the capacity of the underlying model*. We give generalization error bounds for matrix completion and non-linear neural networks, based solely on the expected explicit regularizer and without additional norm constraints on the predictors.

Finally, we note that Mou et al. (2018) give Rademacher complexity bounds that are independent of parameter norms, but require boundedness of the network output. Further, rather than bound generalization gap with a function that vanishes asymptotically with sample size, Mou et al. (2018) bound the *one-sided* gap between population risk and the sum of empirical risk and expected explicit regularizer. We show that for two-layer networks, the expected explicit regularizer is a positive term, implying that generalization error of Mou et al. (2018) does not go to zero, unless the dropout rate goes to zero; see the remark following Corollary 1 for a formal statement. We emphasize that this is *not* the case in successful machine learning systems, as the inventors of Dropout (Srivastava et al., 2014) pointed out "[dropout rate] can be chosen using a validation set or can simply be set at 0.5, which seems to be close to optimal for a wide range of networks and tasks."

There are a bunch of other works that do not fall into any of the categories above, and, in fact, are somewhat unrelated to the focus in this paper. Nonetheless, we discuss them here for completeness. For instance, Gal & Ghahramani (2016) study dropout as Bayesian approximation. Bank & Giryes (2018) draw insights from frame theory to connect the notion of equiangular tight frames with dropout training in auto-encoders. Li et al. (2016) study a variant based on multinomial sampling (different nodes dropped with different rates), and establish sub-optimality bounds for learning linear models (for convex Lipschitz loss functions).

Matrix Factorization with Dropout. Our study of dropout is motivated in part by recent works of Cavazza et al. (2018), Mianjy et al. (2018), and Mianjy & Arora (2019). This line of work was initiated by Cavazza et al. (2018), who studied dropout for low-rank matrix factorization without constraining the rank of the factors or adding an explicit regularizer to the objective. They show that dropout in the context of matrix factorization yields an explicit regularizer whose convex envelope is given by nuclear norm. This result is further strengthened by Mianjy et al. (2018) who show that induced regularizer is indeed nuclear norm.

While matrix factorization is not a learning problem per se (for instance, what is training versus test data), in follow-up works by Mianjy et al. (2018) and Mianjy & Arora (2019), the authors show that training deep linear networks with ℓ_2-loss using dropout reduces to the matrix factorization problem if the marginal distribution of the input feature vectors is assumed to be isotropic, i.e., $\mathbb{E}[\mathrm{x}\mathrm{x}^\top] = \mathrm{I}$. We note that this is a strong assumption. If we do not assume isotropy, we show that dropout induces a data-dependent regularizer which amounts to a simple scaling of the parameters and, therefore, does not control capacity in any meaningful way. We revisit this discussion in Section 4. To summarize, while we are motivated by Cavazza et al. (2018), the problem setup, the nature of statements in this paper, and the tools we use are different from that in Cavazza et al. (2018). Our proofs are simple and quickly verified. We do build closely on the prior work of Mianjy et al. (2018).

However, different from Mianjy et al. (2018), we rigorously argue for dropout in matrix completion by 1) showing that the induced regularizer is equal to weighted trace-norm, which as far as we know, is a novel result, 2) giving strong generalization bounds, and 3) providing extensive experimental evidence that dropout provides state of the art performance on one of the largest datasets in recommendation systems research. Beyond that we rigorously extend our results to two layer ReLU networks, describe the explicit regularizer, bound the Rademacher complexity of the hypothesis class controlled by dropout, show precise generalization bounds, and support them with empirical results.

1.2. Notation and Preliminaries

We denote matrices, vectors, scalar variables and sets by Roman capital letters, Roman small letters, small letters, and script letters, respectively (e.g. X, x, x, and $\mathcal{X}$). For any integer d, we represent the set $\{1, \dots, d\}$ by $[d]$. For any vector $\mathrm{x} \in \mathbb{R}^d$, $\mathrm{diag}(\mathrm{x}) \in \mathbb{R}^{d \times d}$ represents the diagonal matrix with the i^{th} diagonal entry equal to x_i, and $\sqrt{\mathrm{x}}$ is the elementwise squared root of x. Let $\|\mathrm{x}\|$ represent the ℓ_2-norm of vector x, and $\|\mathrm{X}\|$, $\|\mathrm{X}\|_F$, and $\|\mathrm{X}\|_*$ represent the spectral norm, the Frobenius norm, and the nuclear norm of matrix X, respectively. Furthermore, $\|\mathrm{X}\|_{p,q} := \left(\sum_j \left(\sum_i |\mathrm{X}_{i,j}|^p\right)^{q/p}\right)^{1/q}$. Let $\mathrm{X}^\dagger$ denote the Moore-Penrose pseudo-inverse of X. Given a positive definite matrix C, we denote the Mahalonobis norm as $\|\mathrm{x}\|_{\mathrm{C}}^2 = \mathrm{x}^\top \mathrm{C}\mathrm{x}$. For a random variable x that takes values in $\mathcal{X}$, given n i.i.d. samples $\{\mathrm{x}_1, \cdots, \mathrm{x}_n\}$, the empirical average of a function $f : \mathcal{X} \to \mathbb{R}$ is denoted by $\widehat{\mathbb{E}}_i[f(\mathrm{x}_i)] := \frac{1}{n}\sum_{i \in [n]} f(\mathrm{x}_i)$. Furthermore, we denote the second moment of x as $\mathrm{C} := \mathbb{E}[\mathrm{x}\mathrm{x}^\top]$. The standard inner product is represented by $\langle \cdot, \cdot \rangle$, for vectors or matrices, where $\langle \mathrm{X}, \mathrm{X}' \rangle = \mathrm{Tr}(\mathrm{X}^\top \mathrm{X}')$.

We are primarily interested in understanding how dropout controls the capacity of the hypothesis class when using dropout for training. To that end, we consider Rademacher complexity, a sample dependent measure of complexity of a hypothesis class that can directly bound the generalization gap (Bartlett & Mendelson, 2002). Given a sample $\mathcal{S} = \{(\mathrm{x}_1, y_1), \dots, (\mathrm{x}_n, y_n)\}$ of size n, the empirical Rademacher complexity of a function class $\mathcal{F}$ with respect to $\mathcal{S}$, and the expected Rademacher complexity are defined, respectively, as $\mathfrak{R}_{\mathcal{S}}(\mathcal{F}) = \mathbb{E}_\sigma \sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(\mathrm{x}_i)$ and $\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_{\mathrm{x}}[\mathfrak{R}_{\mathcal{S}}(\mathcal{F})]$. where σ_i are i.i.d. Rademacher random variables.

2. Matrix Sensing

We begin with understanding dropout for matrix sensing, a problem which arguably is an important instance of a matrix learning problem with lots of applications, and is well understood from a theoretical perspective. The problem setup is the following. Let $\mathrm{M}_* \in \mathbb{R}^{d_2 \times d_0}$ be a matrix with rank $r_* := \mathrm{Rank}(\mathrm{M}_*)$. Let $\mathrm{A}^{(1)}, \dots, \mathrm{A}^{(n)}$ be a set of measurement matrices of the same size as M_*. The goal of matrix sensing is to recover the matrix M_* from n observations of the form $y_i = \langle \mathrm{M}_*, \mathrm{A}^{(i)} \rangle$ such that $n \ll d_2 d_0$. A natural approach is to represent the matrix in terms of factors and solve the following *empirical risk* minimization problem:

$$\min_{\mathrm{U,V}} \widehat{L}(\mathrm{U,V}) := \widehat{\mathbb{E}}_i(y_i - \langle \mathrm{UV}^\top, \mathrm{A}^{(i)} \rangle)^2 \tag{1}$$

where $\mathrm{U} = [\mathrm{u}_1, \dots, \mathrm{u}_{d_1}] \in \mathbb{R}^{d_2 \times d_1}, \mathrm{V} = [\mathrm{v}_1, \dots, \mathrm{v}_{d_1}] \in \mathbb{R}^{d_0 \times d_1}$. When the number of factors is unconstrained, i.e., when $d_1 \gg r_*$, there exist many "bad" empirical minimizers, i.e., those with a large *true risk* $L(\mathrm{U,V}) := \mathbb{E}(y - \langle \mathrm{UV}^\top, \mathrm{A} \rangle)^2$. Interestingly, Li et al. (2018) showed recently that under a restricted isometry property (RIP), despite the existence of such poor ERM solutions, gradient descent with proper initialization is *implicitly* biased towards finding solutions with minimum nuclear norm – this is an important result which was first conjectured and empirically verified by Gunasekar et al. (2017). We do not make an RIP assumption here. Further, we argue that for the most part, modern machine learning systems employ *explicit* regularization techniques. In fact, as we show in the experimental section, the *implicit* bias due to (stochastic) gradient descent does not prevent it from blatant overfitting in the matrix completion problem.

We propose solving the ERM problem (1) using dropout, where at training time, corresponding columns of U and V are dropped uniformly at random. As opposed to an *implicit* effect of gradient descent, dropout *explicitly* regularizes the empirical objective. It is then natural to ask, in the case of matrix sensing, if dropout also biases the ERM towards certain low norm solutions. To answer this, we begin with the observation that dropout can be viewed as an instance of SGD on the following objective (Wang & Manning, 2013; Srivastava et al., 2014) $\widehat{L}_{\text{drop}}(\mathrm{U,V}) = \widehat{\mathbb{E}}_j \mathbb{E}_{\mathrm{B}}(y_j - \langle \mathrm{UBV}^\top, \mathrm{A}^{(j)} \rangle)^2$, where $\mathrm{B} \in \mathbb{R}^{d_1 \times d_1}$ is a diagonal matrix whose diagonal elements are Bernoulli random variables distributed as $\mathrm{B}_{ii} \sim \frac{1}{1-p}\mathrm{Ber}(1-p)$. It is easy to show that for $p \in [0,1)$:

$$\widehat{L}_{\text{drop}}(\mathrm{U,V}) = \widehat{L}(\mathrm{U,V}) + \frac{p}{1-p}\widehat{R}(\mathrm{U,V}), \tag{2}$$

where $\widehat{R}(\mathrm{U,V}) := \sum_{i=1}^{d_1} \widehat{\mathbb{E}}_j(\mathrm{u}_i^\top \mathrm{A}^{(j)} \mathrm{v}_i)^2$ is a data-dependent term that captures the *explicit* regularizer due to dropout. A similar result was shown by Mianjy et al. (2018), but we provide a proof for completeness (see Proposition 2 in the Appendix).

Furthermore, given that we seek a minimum of $\widehat{L}_{\text{drop}}$, it suffices to consider the factors with the minimal value of the regularizer among all that yield the same empirical loss. This motivates studying the following distribution-dependent *induced* regularizer:

$$\Theta(\mathrm{M}) := \min_{\mathrm{UV}^\top = \mathrm{M}} R(\mathrm{U,V}), \text{ where } R(\mathrm{U,V}) := \mathbb{E}_{\mathrm{A}}[\widehat{R}(\mathrm{U,V})].$$

We instantiate induced regularizer for two instances of random measurements (Prop. 3 in Appendix).

Gaussian Measurements. For all $j \in [n]$, let $\mathrm{A}^{(j)}$ be standard Gaussian matrices. In this case, it is easy to see that $\mathrm{L(U,V)} = \|\mathrm{M}_* - \mathrm{UV}^\top\|_F^2$ and we recover the matrix factorization problem. Furthermore, we know from Mianjy & Arora (2019) that dropout regularizer acts as trace-norm regularization, i.e., $\Theta(\mathrm{M}) = \frac{1}{d_1}\|\mathrm{M}\|_*^2$.

Matrix Completion. For all $j \in [n]$, let $\mathrm{A}^{(j)}$ be an indicator matrix drawn from a product distribution over the rows and columns. That is, the probability of choosing the indicator of the (i,k)-th element is $p(i)q(k)$, where $p(i)$ and $q(k)$ denote the probability of choosing the i-th row and the k-th column, respectively. Then, $\Theta(\mathrm{M}) = \frac{1}{d_1}\|\operatorname{diag}(\sqrt{p})\mathrm{UV}^\top \operatorname{diag}(\sqrt{q})\|_*^2$ is the *weighted trace-norm* studied by Srebro & Salakhutdinov (2010) and Foygel et al. (2011).

These observations are specifically important because they connect dropout, an algorithmic heuristic in deep learning, to strong complexity measures that are empirically effective as well as theoretically well understood. To illustrate, here we give a generalization bound for matrix completion using dropout in terms of the value of the *explicit* regularizer at the minimizer.

Theorem 1. Assume that $d_2 \geq d_0$ and $\|\mathrm{M}_*\| \leq 1$. Furthermore, assume that $\min_{i,k} p(i)q(k) \geq \frac{\log(d_2)}{n\sqrt{d_2 d_0}}$. Let Let $(\mathrm{U,V})$ be the output of ERM with dropout with $R(\mathrm{U,V}) \leq \alpha/d_1$. Then, for any $\delta \in (0,1)$, the following generalization bounds holds with probability at least $1-\delta$ over a sample of size n:

$$L(g(\mathrm{UV}^\top)) \leq \widehat{L}(\mathrm{U,V}) + 8\sqrt{\frac{2\alpha d_2 \log(d_2) + \frac{1}{4}\log(2/\delta)}{n}}$$

where $g(\mathrm{M})$ thresholds M at ±1, i.e. $g(\mathrm{M})(i,j) = \max\{-1, \min\{1, \mathrm{M}(i,j)\}\}$, and $L(g(\mathrm{UV}^\top)) := \mathbb{E}(y - \langle g(\mathrm{UV}^\top), \mathrm{A} \rangle)^2$ is the *true risk* of $g(\mathrm{UV}^\top)$.

The proof of Theorem 1 follows from standard generalization bounds for ℓ_2 loss (Mohri et al., 2018) based on the Rademacher complexity (Bartlett & Mendelson, 2002) of the class of functions with weighted trace-norm bounded by $\sqrt{\alpha}$, i.e. $\mathcal{M}_\alpha := \{\mathrm{M} : \|\operatorname{diag}(\sqrt{\mathrm{p}})\mathrm{M}\operatorname{diag}(\sqrt{\mathrm{q}})\|_*^2 \leq \alpha\}$. The non-degeneracy condition $\min_{i,j} p(i)q(j) \geq \frac{\log(d_2)}{n\sqrt{d_2 d_0}}$ is required to obtain a bound on the Rademacher complexity of $\mathcal{M}_\alpha$, as established by Foygel et al. (2011). Furthermore, since the induced regularizer is scaled as $1/d_1$ compared to the squared weighted trace-norm, i.e. $\Theta(\mathrm{UV}^\top) = \frac{1}{d_1}\|\operatorname{diag}(\sqrt{\mathrm{p}})\mathrm{UV}^\top \operatorname{diag}(\sqrt{\mathrm{q}})\|_*^2$, we scale α accordingly by letting $R(\mathrm{U,V}) \leq \alpha/d_1$.

In practice, for models that are trained with dropout, the training error $\widehat{L}(\mathrm{U,V})$ is negligible (see Figure 1 for experiments on the MovieLens dataset). Moreover, given that the sample size is large enough, the third term can be made arbitrarily small. Having said that, the second term, which is $\tilde{O}(\sqrt{\alpha d_2/n})$, dominates the right hand side of generalization error bound in Theorem 9. In Appendix, we also give optimistic generalization bounds that decay as $\tilde{O}(\alpha d_2/n)$.

Finally, the required sample size depends on the value of the explicit regularizer (i.e., α/d_1), and hence, on the dropout rate p. In particular, increasing the dropout rate increases the regularization parameter $\lambda := \frac{p}{1-p}$, thereby intensifying the penalty due to the explicit regularizer. Intuitively, a larger dropout rate p results in a smaller α, thereby a tighter generalization gap can be guaranteed. We show through experiments that that is indeed the case in practice.

3. Non-linear Networks

Next, we focus on neural networks with a single hidden layer. Let $\mathcal{X} \subseteq \mathbb{R}^{d_0}$ and $\mathcal{Y} \subseteq [-1,1]^{d_2}$ denote the input and output spaces, respectively. Let $\mathcal{D}$ denote the joint probability distribution on $\mathcal{X} \times \mathcal{Y}$. Given n examples $\{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n \sim \mathcal{D}^n$ drawn i.i.d. from the joint distribution and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the goal of learning is to find a hypothesis $f_\mathrm{w} : \mathcal{X} \to \mathcal{Y}$, parameterized by w, that has a small *population risk* $L(f_\mathrm{w}) := \mathbb{E}_\mathcal{D}[\ell(f_\mathrm{w}(\mathrm{x}), \mathrm{y})]$.

We focus on the squared ℓ_2 loss, i.e., $\ell(\mathrm{y}, \mathrm{y}') = \|\mathrm{y} - \mathrm{y}'\|^2$, and study the generalization properties of the dropout algorithm for minimizing the *empirical risk* $\widehat{L}(f_\mathrm{w}) := \widehat{\mathbb{E}}_i[\|\mathrm{y}_i - f_\mathrm{w}(\mathrm{x}_i)\|^2]$. We consider the hypothesis class associated with feed-forward neural networks with 2 layers, i.e., functions of the form $f_\mathrm{w}(\mathrm{x}) = \mathrm{U}\sigma(\mathrm{V}^\top \mathrm{x})$, where $\mathrm{U} = [\mathrm{u}_1, \ldots, \mathrm{u}_{d_1}] \in \mathbb{R}^{d_2 \times d_1}, \mathrm{V} = [\mathrm{v}_1, \ldots, \mathrm{v}_{d_1}] \in \mathbb{R}^{d_0 \times d_1}$ are the weight matrices. The parameter w is the collection of weight matrices $\{\mathrm{U}, \mathrm{V}\}$ and $\sigma : \mathbb{R} \to \mathbb{R}$ is the ReLU activation function applied entrywise to an input vector. As in Section 2, we view dropout as an instance of stochastic gradient descent on the following *dropout objective*:

$$\widehat{L}_{\text{drop}}(\mathrm{w}) := \widehat{\mathbb{E}}_i \mathbb{E}_\mathrm{B} \|\mathrm{y}_i - \mathrm{U}\mathrm{B}\sigma(\mathrm{V}^\top \mathrm{x}_i)\|^2, \tag{3}$$

where B is a diagonal random matrix with diagonal elements distributed i.i.d. as $\mathrm{B}_{ii} \sim \frac{1}{1-p}\mathrm{Bern}(1-p)$, $i \in [d_1]$, for some *dropout rate* p. We seek to understand the *explicit regularizer* due to dropout:

$$\widehat{R}(\mathrm{w}) := \widehat{L}_{\text{drop}}(\mathrm{w}) - \widehat{L}(f_\mathrm{w}). \tag{4}$$

We denote the output of the i-th hidden node on an input vector x by $a_i(\mathrm{x}) \in \mathbb{R}$; for example, $a_2(\mathrm{x}) = \sigma(\mathrm{v}_2^\top \mathrm{x})$. Similarly, the vector $\mathrm{a}(\mathrm{x}) \in \mathbb{R}^{d_1}$ denotes the activation of the hidden layer on input x. Using this notation, we can rewrite the objective in (3) as $\widehat{L}_{\text{drop}}(\mathrm{w}) := \mathbb{E}_i \mathbb{E}_\mathrm{B} \|\mathrm{y}_i - \mathrm{U}\mathrm{B}\mathrm{a}(\mathrm{x}_i)\|^2$. It is then easy to show that the regularizer due to dropout in (4) is given as (Proposition 4 in Appendix):

$$\widehat{R}(\mathrm{w}) = \frac{p}{1-p} \sum_{j=1}^{d_1} \|\mathrm{u}_j\|^2 \widehat{a}_j^2, \text{ where } \widehat{a}_j = \sqrt{\widehat{\mathbb{E}}_i a_j(\mathrm{x}_i)^2}.$$

The explicit regularizer $\widehat{R}(\mathrm{w})$ is a summation over hidden nodes, of the product of the squared norm of the outgoing weights with the empirical second moment of the output of the corresponding neuron. We should view it as a data-dependent variant of the ℓ_2 path-norm of the network, studied recently by Neyshabur et al. (2015b) and shown to yield capacity control in deep learning. Indeed, if we consider ReLU activations and input distributions that are symmetric and isotropic (Mianjy et al., 2018), the expected regularizer is equal to the sum over all paths from input to output of the product of the squares of weights along the paths, i.e., $R(\mathrm{w}) := \mathbb{E}[\widehat{R}(\mathrm{w})] = \frac{1}{2} \sum_{i_0, i_1, i_2 = 1}^{d_0, d_1, d_2} \mathrm{U}(i_2, i_1)^2 \mathrm{V}(i_0, i_1)^2$, which is precisely the squared ℓ_2 path-norm of the network. We refer the reader to Proposition 5 in the Appendix for a formal statement and proof.

Generalization Bounds. To understand the generalization properties of dropout, we focus on the following distribution-dependent hypothesis class

$$\mathcal{F}_\alpha := \{f_\mathrm{w} : \mathrm{x} \mapsto \mathrm{u}^\top \sigma(\mathrm{V}^\top \mathrm{x}), \sum_{i=1}^{d_1} |u_i| a_i \leq \alpha\}, \tag{5}$$

where $\mathrm{u} \in \mathbb{R}^{d_1}$ is the top layer weight vector, u_i denotes the i-th entry of u, and $a_i^2 := \mathbb{E}_\mathrm{x}[\widehat{a}_i^2] = \mathbb{E}_\mathrm{x}[a_i(\mathrm{x})^2]$ is the expected squared activation of the i-th hidden node. For simplicity, we focus on networks with one output neuron ($d_2 = 1$); extension to multiple output neurons is straightforward.

We argue that networks trained with dropout belong to the class $\mathcal{F}_\alpha$, for a small value of α. In particular, by Cauchy-Schwartz inequality, it is easy to to see that $\sum_{i=1}^{d_1} |u_i| a_i \leq \sqrt{d_1 R(\mathrm{w})}$. Thus, for a fixed width, dropout implicitly controls the function class $\mathcal{F}_\alpha$. More importantly, this inequality is loose if a small subset of hidden nodes $\mathcal{J} \subset [d_1]$ *"co-adapt"* in a way that for all $j \in [d_1] \setminus \mathcal{J}$, the other hidden nodes are almost inactive, i.e. $u_j a_j \approx 0$. In other words, by minimizing the expected regularizer, dropout is biased towards networks where gap between $R(\mathrm{w})$ and $(\sum_{i=1}^{d_1} |u_i| a_i)^2 / d_1$ is small, which in turn happens if $|u_i| a_i \approx |u_j| a_j, \forall i, j \in [d_1]$. In this sense, dropout breaks *"co-adaptation"* between neurons by promoting solutions with nearly equal contribution from hidden neurons.

As we mentioned in the introduction, a bound on the dropout regularizer is not sufficient to guarantee a bound on a norm-based complexity measures that are common in the deep learning literature (see, e.g., Golowich et al. (2018) and the references therein), whereas a norm bound on the weight vector would imply a bound on the explicit regularizer due to dropout. Formally, we show the following.

Proposition 1. *For any $C > 0$, there exists a distribution on the unit Euclidean sphere, and a network $f_\mathrm{w} : \mathrm{x} \mapsto \sigma(\mathrm{w}^\top \mathrm{x})$, such that $R(\mathrm{w}) = \sqrt{\mathbb{E}\sigma(\mathrm{w}^\top \mathrm{x})^2} \leq 1$, while $\|\mathrm{w}\| > C$.*

In other words, even though we connect the dropout regularizer to path-norm, the data-dependent nature of the regularizer prevents us from leveraging that connection in data-independent manner (i.e., for all distributions). At the same time, making strong distributional assumptions (as in Proposition 5) would be impractical. Instead, we argue for the following milder condition on the input distribution which we show as sufficient to ensure generalization.

Assumption 1 (β**-retentive).** The marginal input distribution is β-retentive for some $\beta \in (0, 1/2]$, if for any non-zero vector $\mathrm{v} \in \mathbb{R}^d$, it holds that $\mathbb{E}\sigma(\mathrm{v}^\top \mathrm{x})^2 \geq \beta \mathbb{E}(\mathrm{v}^\top \mathrm{x})^2$.

Intuitively, what the assumption implies is that the variance (aka, the information or signal in the data) in the pre-activation at any node in the network is not quashed considerably due to the non-linearity. In fact, no reasonable training algorithm should learn weights where β is small. However, we steer clear from algorithmic aspects of dropout training, and make the assumption above for every weight vector as we need to take a union bound. We now present the first main result of this section, which bounds the Rademacher complexity of $\mathcal{F}_\alpha$ in terms of α, the retentiveness coefficient β, and Mahalanobis norm of data w.r.t. the pseudo-inverse of the second moment matrix, i.e. $\|\mathrm{X}\|_{\mathrm{C}^\dagger}^2 = \sum_{i=1}^n \mathrm{x}_i^\top \mathrm{C}^\dagger \mathrm{x}_i$.

Theorem 2. *For any sample* $S = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n$ *of size* n, $\mathfrak{R}_S(\mathcal{F}_\alpha) \leq \frac{2\alpha\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{n\sqrt{\beta}}$. *Furthermore, it holds for the expected Rademacher complexity that* $\mathfrak{R}_n(\mathcal{F}_\alpha) \leq 2\alpha\sqrt{\frac{\mathrm{Rank}(\mathrm{C})}{\beta n}}$.

First, note that the bound depends on the quantity $\|\mathrm{X}\|_{\mathrm{C}^\dagger}$ which can be in the same order as $\|\mathrm{X}\|_F$ with both scaling as $\asymp \sqrt{nd_0}$; the latter is more common in the literature (Neyshabur et al., 2018; Bartlett et al., 2017; Neyshabur et al., 2017; Golowich et al., 2018; Neyshabur et al., 2015b). This is unfortunately unavoidable, unless one makes stronger distributional assumptions.

Second, as we discussed earlier, the dropout regularizer directly controls the value of α, thereby controlling the Rademacher complexity in Theorem 2. This bound also gives us a bound on the Rademacher complexity of the networks trained using dropout. To see that, consider the following class of networks with bounded explicit regularizer, i.e., $\mathcal{H}_r := \{h_\mathrm{w} : \mathrm{x} \mapsto \mathrm{u}^\top \sigma(\mathrm{V}^\top \mathrm{x}),\ R(\mathrm{u}, \mathrm{V}) \leq r\}$. Then, Theorem 2 yields $\mathfrak{R}_S(\mathcal{H}_r) \leq \frac{2\sqrt{d_1 r}\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{n\sqrt{\beta}}$. In fact, we can show that this bound is tight up to $1/\sqrt{\beta}$ by a reduction to the linear case. Formally, we show the following.

Theorem 3 (Lowerbound)**.** *There is a constant c such that for any* $r > 0$, $\mathfrak{R}_S(\mathcal{H}_r) \geq \frac{c\sqrt{d_1 r}\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{n}$.

Moreover, it is easy to give a generalization bound based on Theorem 2 that depends only on the distribution dependent quantities α and β. Let $g_\mathrm{w}(\cdot) := \max\{-1, \min\{1, f_\mathrm{w}(\cdot)\}\}$ project the network output f_w onto the range $[-1, 1]$. We have the following generalization gurantees for g_w.

Corollary 1. *For any* $\mathrm{w} \in \mathcal{F}_\alpha$, *for any* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$ *over a sample* $\mathcal{S}$ *of size* n, *we have* $L(g_\mathrm{w}) \leq \widehat{L}(g_\mathrm{w}) + \frac{16\alpha\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{\sqrt{\beta}n} + 12\sqrt{\frac{\log(2/\delta)}{2n}}$.

Comparison with Mou et al. (2018) We note that the Corollary above bounds the generalization gap, i.e., $L(\cdot) - \widehat{L}(\cdot)$. In contrast, Mou et al. (2018) bound $L(\cdot) - \widehat{L}_{\text{drop}}(\cdot)$, where $\widehat{L}_{\text{drop}}(\mathrm{w}) = \widehat{L}(f_\mathrm{w}) + \widehat{R}(\mathrm{w})$, as in Equation (4). The explicit regularizer $\widehat{R}(\cdot)$ is a positive quantity that does not vanish with the sample size. Therefore, the bound of Mou et al. (2018) can guarantee that the generalization gap decays as $1/\sqrt{n}$ only if the dropout rate decreases as $1/\sqrt{n}$ (to ensure that $\widehat{R}(\cdot) = O(1/\sqrt{n})$). In sharp contrast, our analysis is valid for *any* dropout rate.

β**-independent Bounds.** Geometrically, β-retentiveness requires that for any hyperplane passing through the origin, both halfspaces contribute significantly to the second moment of the data in the direction of the normal vector. It is not clear, however, if β can be estimated efficiently on a dataset. Nonetheless, when $\mathcal{X} \subseteq \mathbb{R}_+^{d_0}$, which is the case for image datasets, a simple *symmetrization* technique, described below, allows us to give bounds that are β-independent. We propose the following randomized symmetrization. *Given a training sample* $\mathcal{S} = \{(\mathrm{x}_i, y_i), i \in [n]\}$, *consider the randomly perturbed dataset,* $\mathcal{S}' = \{(\zeta_i \mathrm{x}_i, y_i),\ i \in [n]\}$, *where* ζ_i*'s are i.i.d. Rademacher random variables.* We give a generalization bound (w.r.t. the original data distribution) for the hypothesis class with bounded regularizer w.r.t. perturbed data distribution.

Corollary 2. *Given an i.i.d. sample* $\mathcal{S} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^n$, *let* $\mathcal{F}'_\alpha := \{f_\mathrm{w} : \mathrm{x} \mapsto u^\top \sigma(V^\top \mathrm{x}),\ \sum_{i=1}^{d_1} |u_i| a'_i \leq \alpha\}$, *where* ${a'_i}^2 := \mathbb{E}_{\mathrm{x},\zeta}[a_i(\zeta \mathrm{x})^2]$. *For any* $\mathrm{w} \in \mathcal{F}'_\alpha$, *for any* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$ *over a sample of size* n *and the randomization in symmetrization, we have that* $L(g_\mathrm{w}) \leq 2\widehat{L}(g_\mathrm{w}) + \frac{46\alpha\|\mathrm{X}\|_{\mathrm{C}^\dagger}}{n} + 24\sqrt{\frac{\log(2/\delta)}{2n}}$, *where* $\widehat{L}$ *is evaluated on the symmetrized sample* $\mathcal{S}'$.

Note that the population risk of the clipped predictor $g_\mathrm{w}(\cdot) := \max\{-1, \min\{1, f_\mathrm{w}(\cdot)\}\}$ is bounded in terms of empirical risk on $\mathcal{S}'$. Finally, we verify in Section 5 that symmetrization of the training set, on MNIST and Fashion-MNIST datasets, does not have an effect on performance of the trained models.

4. Role of Parametrization

In this section, we argue that parametrization plays an important role in determining the nature of the inductive bias. We begin by considering matrix sensing in non-factorized form, which entails minimizing $\widehat{L}(\mathrm{M}) := \widehat{\mathbb{E}}_i(y_i - \langle \mathrm{vec}\,(\mathrm{M}), \mathrm{vec}(\mathrm{A}^{(i)})\rangle)^2$, where $\mathrm{vec}\,(\mathrm{M})$ denotes the column vectorization of M. Then, the expected explicit regularizer due to dropout equals $R(\mathrm{M}) = \frac{p}{1-p}\|\,\mathrm{vec}\,(\mathrm{M})\,\|_{\mathrm{diag}(\mathrm{C})}^2$,

	plain SGD		dropout			
width	last iterate	best iterate	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$
$d_1 = 30$	0.8041	0.7938	0.7805	0.785	0.7991	0.8186
$d_1 = 70$	0.8315	0.7897	0.7899	0.7771	0.7763	0.7833
$d_1 = 110$	0.8431	0.7873	0.7988	0.7813	0.7742	0.7743
$d_1 = 150$	0.8472	0.7858	0.8042	0.7852	0.7756	0.7722
$d_1 = 190$	0.8473	0.7844	0.8069	0.7879	0.7772	0.772

Table 1. MovieLens dataset: Test RMSE of plain SGD as well as the dropout algorithm with various dropout rates for various factorization sizes. The grey cells shows the best performance(s) in each row.

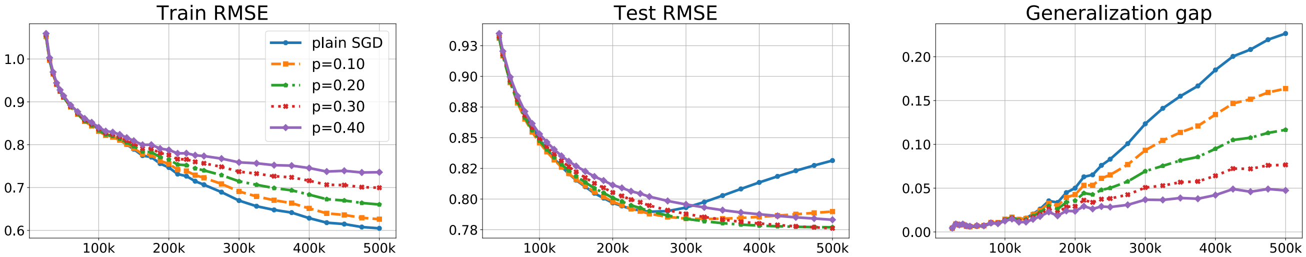

Figure 1. MovieLens dataset: training error (**left**), test error (**middle**), and generalization gap (**right**) for plain SGD and dropout with $p \in \{0.1, 0.2, 0.3, 0.4\}$ as a function of number of iterations; factorization size, $d_1 = 70$.

where $\mathrm{C} = \mathbb{E}[\mathrm{vec}\left(\mathrm{A}\right)\mathrm{vec}\left(\mathrm{A}\right)^\top]$ is the second moment of the measurement matrices. For instance, with Gaussian measurements, the second moment equals the identity matrix, in which case, the regularizer reduces to the Frobenius norm of the parameters $R(\mathrm{M}) = \frac{p}{1-p}\|\mathrm{M}\|_F^2$. While such a ridge penalty yields a useful inductive bias in linear regression, it is not *"rich"* enough to capture the kind of inductive bias that provides rank control in matrix sensing.

However, simply representing the hypotheses in a factored form alone is not sufficient in terms of imparting a rich inductive bias to the learning problem. Recall that in linear regression, dropout, when applied on the input features, yields ridge regularization. However, if we were to represent the linear predictor in terms of a deep linear network, then we argue that the effect of dropout is markedly different. Consider a deep linear network, $f_{\mathrm{w}} : \mathrm{x} \mapsto \mathrm{W}_k \cdots \mathrm{W}_1 \mathrm{x}$ with a single output neuron. In this case, Mianjy & Arora (2019) show that $\nu\|f\|_{\mathrm{C}}^2 = \min_{f_{\mathrm{w}}=f} \widehat{R}(\mathrm{w})$, where ν is a regularization parameter independent of the parameters w. Consequently, in deep linear networks with a single output neuron, dropout reduces to solving

$$\min_{\mathrm{u}\in\mathbb{R}^{d_0}} \widehat{\mathbb{E}}_i(y_i - \mathrm{u}^\top \mathrm{x}_i)^2 + \nu\|\mathrm{u}\|_{\mathrm{C}}^2.$$

All the minimizers of the above problem are solutions to the system of linear equations $(1 + \frac{\nu}{n})\mathrm{X}\mathrm{X}^\top\mathrm{u} = \mathrm{X}\mathrm{y}$, where $\mathrm{X} = [\mathrm{x}_1, \ldots, \mathrm{x}_n] \in \mathbb{R}^{d_0 \times n}, \mathrm{y} = [y_1; \ldots; y_n] \in \mathbb{R}^n$ are the design matrix and the response vector, respectively. In other words, the dropout regularizer manifests itself merely as a scaling of the parameters.

What we argue above may at first seem to contradict the results of Section 2 on matrix sensing, which is arguably an instance of regression with a two-layer linear network. Note though that casting matrix sensing in a factored form as a linear regression problem requires us to use a convolutional structure. This is easy to check since

$$\begin{aligned}\langle \mathrm{U}\mathrm{V}^\top, \mathrm{A}\rangle &= \langle \mathrm{vec}\left(\mathrm{U}^\top\right), \mathrm{vec}\left(\mathrm{V}^\top\mathrm{A}^\top\right)\rangle \\ &= \langle \mathrm{vec}\left(\mathrm{U}^\top\right), (\mathrm{I}_{d_2} \otimes \mathrm{V}^\top)\,\mathrm{vec}\left(\mathrm{A}^\top\right)\rangle,\end{aligned}$$

where $\otimes$ is the Kronecker product, and we used the fact that $\mathrm{vec}\left(\mathrm{A}\mathrm{B}\right) = (\mathrm{I} \otimes \mathrm{A})\,\mathrm{vec}\left(\mathrm{B}\right)$ for any pair of matrices A, B. The expression $(\mathrm{I} \otimes \mathrm{V}^\top)$ represents a fully connected convolutional layer with d_1 filters specified by columns of V. The convolutional structure in addition to dropout is what imparts the problem of matrix sensing the nuclear norm regularization. For nonlinear networks, however, a simple feed-forward structure suffices as we saw in Section 3.

5. Experimental Results

In this section, we report our empirical findings on real world datasets. All results are averaged over 50 independent runs with random initialization.

Matrix Completion. We evaluate dropout on the MovieLens dataset (Harper & Konstan, 2016), a publicly available collaborative filtering dataset that contains 10M ratings for 11K movies by 72K users of the online movie recommender service MovieLens. We initialize the factors using the standard He initialization scheme (He et al., 2015). We train the model for 100 epochs over the training data, where we use a fixed learning rate of $\texttt{lr} = 1$, and a batch size of 2000. We report the results for plain SGD ($p = 0.0$) as well as the dropout algorithm with $p \in \{0.1, 0.2, 0.3, 0.4\}$.

Figure 1 shows the progress in terms of the training and test error as well as the gap between them as a function

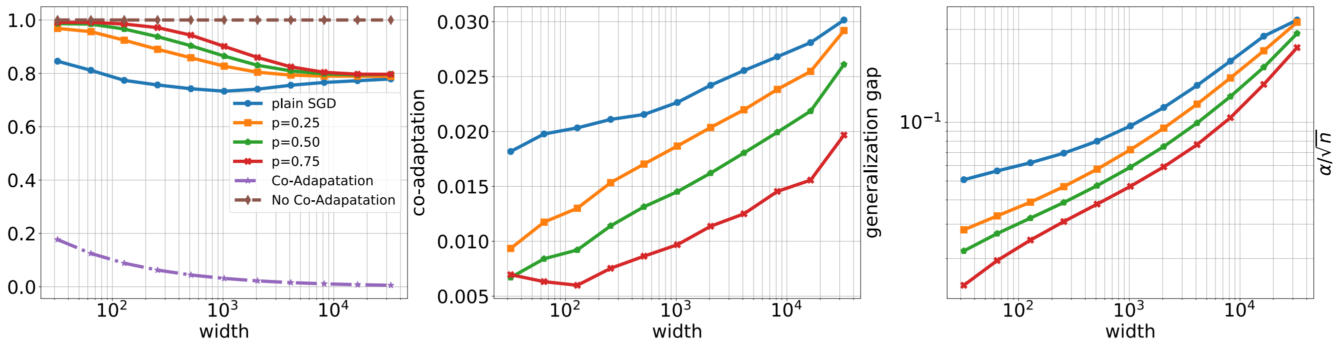

Figure 2. (**left**) *"co-adaptation"*; (**middle**) generalization gap; and (**right**) $\alpha/\sqrt{n}$ as a function of the width of networks trained with dropout on MNIST. In left figure, the dashed brown and dotted purple lines represent minimal and maximal co-adaptations, respectively.

of the number of iterations for $d_1 = 70$. It can be seen that plain SGD is the fastest in minimizing the empirical risk. The dropout rate clearly determines the trade-off between the goodness of fit and the model complexity: as the dropout rate p increases, the algorithm favors less complex solutions that suffer larger empirical error (left figure) but enjoy smaller generalization gap (right figure). The best trade-off here seems to be achieved by a moderate dropout rate of $p = 0.3$. We observe similar behaviour for different factorization sizes; please see the Appendix for additional plots with factorization sizes $d_1 \in \{30, 110, 150, 190\}$.

It is remarkable, how even in the "simple" problem of matrix completion, plain SGD lacks a proper inductive bias. As it is clearly depicted in the middle plot, without *explicit* regularization – in particular early stopping or dropout in this figure – SGD suffers from gross overfitting. We further illustrate this fact in Table 1, where we compare the test root-mean-squared-error (RMSE) of plain SGD with the dropout algorithm, for various factorization sizes. To show the superiority of dropout over SGD with early stopping, we give SGD the advantage of having access to the *test set* (and not a separate validation set), and report the best iterate in the third column. Even with this impractical privilege, dropout performs significantly better (> 0.01 difference in test RMSE).

Neural Networks. We train 2-layer neural networks with and without dropout, on MNIST dataset of handwritten digits and Fashion MNIST dataset of Zalando's article images, each of which contains 60K training examples and 10K test examples, where each example is a 28×28 grayscale image, associated with a label from 10 classes. We extract two classes $\{4, 7\}$ and label them as $\{-1, +1\}$. We observe similar results across other choices of target classes. The learning rate in all experiments is set to $\mathtt{lr} = 1e - 3$. We train the models for 30 epochs over the training set. We run the experiments *both with and without symmetrization*. Here we only report the results with symmetrization, and on the MNIST dataset. For experiments without symmetrization, and experiments on FashionMNIST, please see the Appendix. We remark that *under the above experimental setting, trained networks achieve* 100% *training accuracy.*

For any node $i \in [d_1]$, define its *flow* as $\psi_i := |u_i|a_i$ (respectively $\psi_i := |u_i|a'_i$ for symmetrized data), which measures the overall contribution of a node to the output of the network. Co-adaptation occurs when a small subset of nodes dominate the overall function of the network. We argue that $\phi(\mathrm{w}) = \frac{\|\psi\|_1}{\sqrt{d_1}\|\psi\|_2}$ is a suitable measure of co-adaptation (or lack thereof) in a network parameterized by w. In case of high co-adaptation, only a few nodes have a high flow, which implies $\phi(\mathrm{w}) \approx \frac{1}{\sqrt{d_1}}$. At the other end of the spectrum, all nodes are equally active, in which case $\phi(\mathrm{w}) \approx 1$. Figure 2 (left) illustrates this measure as a function of the network width for several dropout rates $p \in \{0, 0.25, 0.5, 0.75\}$. In particular, we observe that a higher dropout rate corresponds to less co-adaptation. More interestingly, even plain SGD is *implicitly* biased towards networks with less co-adaptation. Moreover, for a fixed dropout rate, the regularization effect due to dropout decreases as we increase the width. Thus, it is natural to expect more co-adaptation as the network becomes wider, which is what we observe in the plots.

The generalization gap is plotted in Figure 2 (middle). As expected, increasing dropout rate decreases the generalization gap. In our experiments, the generalization gap increases with the width of the network. The figure on the right shows the quantity $\alpha/\sqrt{n}$ that shows up in the Rademacher complexity bounds in Section 3. We note that, the bound on the Rademacher complexity is predictive of the generalization gap, in the sense that a smaller bound corresponds to a curve with smaller generalization gap.

6. Conclusion

In this paper, we studied the capacity control provided by dropout in matrix completion as well as two-layer neural networks. The focus here has been on *understanding* how the expected explicit regularizer alone – withought any additional norm-bounds on the weights – can provide generalization. If one is interested in *predicting* the generalization gap, then one can estimate the (empirical) explicit regularizer on a held-out dataset, and appeal to simple concentration arguments, just as we do in our experiments.

Acknowledgements

This research was supported, in part, by NSF BIGDATA award IIS-1546482 and NSF CAREER award IIS-1943251. The seeds of this work were sown during the summer 2019 workshop on the Foundations of Deep Learning at the Simons Institute for the Theory of Computing. Raman Arora acknowledges the support provided by the Institute for Advanced Study, Princeton, New Jersey as part of the special year on Optimization, Statistics, and Theoretical Machine Learning.

References

Baldi, P. and Sadowski, P. J. Understanding dropout. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2814–2822, 2013.

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

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6240–6249, 2017.

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.

Chen, D. and Manning, C. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750, 2014.

Dahl, G. E., Sainath, T. N., and Hinton, G. E. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 8609–8613. IEEE, 2013.

Foygel, R., Shamir, O., Srebro, N., and Salakhutdinov, R. R. Learning with the weighted trace-norm under arbitrary sampling distributions. In *Advances in Neural Information Processing Systems*, pp. 2133–2141, 2011.

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.

Golowich, N., Rakhlin, A., and Shamir, O. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pp. 297–299, 2018.

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.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19, 2016.

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M., and Larochelle, H. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

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.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

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.

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.

Li, Z., Gong, B., and Yang, T. Improved dropout for shallow and deep learning. In *Advances in neural information processing systems*, pp. 2523–2531, 2016.

McAllester, D. A PAC-bayesian tutorial with a dropout bound. *arXiv preprint arXiv:1307.2118*, 2013.

Mianjy, P. and Arora, R. On dropout and nuclear norm regularization. In *International Conference on Machine Learning*, 2019.

Mianjy, P. and Arora, R. On convergence and generalization of dropout training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21151–21161, 2020.

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

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 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.

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.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Pham, V., Bluche, T., Kermorvant, C., and Louradour, J. Dropout improves recurrent neural networks for handwriting recognition. In *2014 14th international conference on frontiers in handwriting recognition*, pp. 285–290. IEEE, 2014.

Srebro, N. and Salakhutdinov, R. R. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*, pp. 2056–2064, 2010.

Srebro, N., Sridharan, K., and Tewari, A. Optimistic rates for learning with a smooth loss. *arXiv preprint arXiv:1009.3896*, 2010.

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

Wager, S., Wang, S., and Liang, P. S. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 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.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pp. 1058–1066, 2013.

Wang, S. and Manning, C. Fast dropout training. In *international conference on machine learning*, pp. 118–126, 2013.

Wei, C., Kakade, S., and Ma, T. The implicit and explicit regularization effects of dropout. *arXiv preprint arXiv:2002.12915*, 2020.

Yang, Z., He, X., Gao, J., Deng, L., and Smola, A. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 21–29, 2016.

Zhai, K. and Wang, H. Adaptive dropout with rademacher complexity regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1uxsye0Z.

Zhang, L., Deng, Z., Kawaguchi, K., Ghorbani, A., and Zou, J. How does mixup help with robustness and generalization? In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=8yKEo06dKNo.