Dropout Training, Data-dependent Regularization, and Generalization Bounds

Wenlong Mou [1] Yuchen Zhou [2] Jun Gao [3] Liwei Wang [3 4]

Abstract

We study the problem of generalization guarantees for dropout training. A general framework is first proposed for learning procedures with random perturbation on model parameters. The generalization error is bounded by sum of two offset Rademacher complexities: the main term is Rademacher complexity of the hypothesis class with minus offset induced by the perturbation variance, which characterizes data-dependent regularization by the random perturbation; the auxiliary term is offset Rademacher complexity for the variance class, controlling the degree to which this regularization effect can be weakened. For neural networks, we estimate upper and lower bounds for the variance induced by truthful dropout, a variant of dropout that we propose to ensure unbiased output and fit into our framework, and the variance bounds exhibits connection to adaptive regularization methods. By applying our framework to ReLU networks with one hidden layer, a generalization upper bound is derived with no assumptions on the parameter norms or data distribution, with $O(1/n)$ fast rate and adaptivity to geometry of data points being achieved at the same time.

1. Introduction

The past six years witnessed the scintillating success of dropout training method in deep learning, ever since the seminal work by (Hinton et al., 2012). It has become a common technique among deep learning practitioners, to randomly mask out part of network during training, and use the entire network for prediction. A series of comprehensive experimental studies have shown the regularization effect incurred by dropout, as the gap between training loss and testing loss is significantly reduced. This simple technique has developed into an active research area, with several variants being proposed, advancing the state-of-the-art test performances in deep learning (Ma et al., 2016; Huang et al., 2016; Goodfellow et al., 2013; Wan et al., 2013; Rippel et al., 2014). Perhaps surprisingly, dropout training usually achieves the better testing performance in the tradeoff between capacity and generalization, compared with standard ℓ^2 regularization. This well-known phenomena was characterized as ”Altitude Training” (Wager et al., 2014): analogous to athletes who were trained in a harder situation than they compete in, the difficulties induced by perturbation during training endow the model with more robustness, making life easier for it when faced with testing data.

Despite its empirical success, current theoretical understanding into this technique is very vague. In their original paper, Hinton et al. (2012) describes dropout training as trying to ensemble exponentially many sub-networks with sharing parameters. However, the final model is actually taking average instead of ensemble. Thus their original argument is weakened by the well-known non-convexity of neural networks' loss functions, as averaged parameter can lead to significantly larger loss. Moreover, as the way in which dropout helps generalization remaining a myth, practitioners usually find themselves confused with different situations where dropout work or not. For example, Ioffe & Szegedy (2015) discovered that dropout doesn't help if ”Batch Normalization (BN)” is used in the network. This was verified by extensive experiments, and equivocally explained as ”some regularization effect” of BN. The theoretical results proposed in this paper, on the other hand, is able to address this issue in a solid and quantitative way.

Wide applications of dropout techniques also motivates a lot of theoretical studies ranging from statistical to computational perspectives. Wager et al. (2013) showed the adaptive regularization effect of dropout training for generalized linear models. They also established connections to adaptive gradient methods such as AdaGrad (Duchi et al., 2011). In (Wager et al., 2014), faster rate convergence was shown to be achieved using dropout, for a class of topic models. For the online learning setting, dropout was considered a kind of perturbation in Follow the Perturbed Leader (FTPL)

[1] Department of EECS, University of California, Berkeley [2] Department of Statistics, University of Wisconsin, Madison [3] Key Laboratory of Machine Perception, MOE, School of EECS, Peking University [4] Center for Data Science, Peking University, Beijing Institute of Big Data Research. Correspondence to: Liwei Wang <wanglw@cis.pku.edu.cn>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

algorithms, and regret bounds free from parameters was shown in (Van Erven & Kotl, 2014). However, all of above addresses the problem essentially in linear models. For deeper models, (Gal & Ghahramani, 2016) gives a Bayesian interpretation for dropout. Several attempts have also been made to study the effect of dropout on the generalization performance of deep neural networks, such as (Gao & Zhou, 2016; Wan et al., 2013). They proved generalization bounds with dropout based on norm assumptions. If dropout really helps generalization in a way different from ℓ_2 regularization (as is known by practitioners and theorists (Helmbold & Long, 2015)), sharper bounds need to be developed even without norm assumptions.

Recently, there are a few research works initiating towards understanding of dropout in multi-layer models. Helmbold & Long (2015) studies inductive bias, i.e., suitable distributions that can be learned via dropout, from computational learning theory perspectives. They also discovered some interesting properties of regularization term induced by dropout, especially their differences with classical ℓ_2 regularization, in their later work (Helmbold & Long, 2017). Their examples provides important insights, though uniform generalization guarantees are not provided. The central question of how dropout training controls the excess risks of learning algorithm remains open, which is essential from both theoretical and practical point of view.

In this paper, we will focus on generalization error bounds of dropout training, and in general the training algorithms with random perturbations that induces data-dependent regularization. To capture the ”altitude training” phenomenon in a distribution-free way, we compare population risk against expected training loss under random perturbation. Both general framework and results for deep neural networks are presented. We summarize our contribution as follows.

1.1. Our Contribution

The contribution made by this paper contains both general framework and specific analysis of neural networks.

1. In Section 3, we propose a distribution-free framework for generalization performance of learning algorithms which randomly perturb the model parameter during training. Assuming the unbiasedness of random perturbation and squared loss function, the generalization performance can be controlled by sum of two offset Rademacher complexity terms (see Theorem 2): for the main term, we obtain a minus quadratic term depending on the variance of random perturbation, in addition to standard form of Rademacher complexity; the second one, which comes from a more technical reason, characterizes how much the regularization effect of variance can be weakened by overfitting.

2. In Section 4, we begin the analysis for deep non-linear neural networks. In order to apply our framework, we first propose a truthful dropout algorithm to make the output unbiased. The variance of truthful dropout is studied with upper and lower bounds presented (see Theorem 4 and 5). An interesting discovery is that, the variance as a regularizer exhibits connections to data-dependent regularization methods such as AdaGrad and batch normalization. We also carry out experiments to show the effectiveness of truthful dropout algorithm, which performs comparable to the vanilla version of dropout training (see Section 6).

3. In Section 5, we utilize the framework with the variance estimates in Section 4, and prove concrete generalization error bounds for a class of one-hidden-layer ReLU neural networks. We obtain a bound assuming only boundedness of output without any norm-based assumptions. The generalization bounds we obtained achieve the $O(1/n)$ fast rate without knowing anything about global optimal point, even though localization arguments are usually unavailable for neural networks. This is due to the self-modulating properties of the minus quadratic term induced by variance offset. The fast rate shows the ”altitude training” effect in non-linear neural network models, without any distribution assumptions. Furthermore, the generalization bound by this data-dependent regularization is also adaptive to geometric properties of data points. It degenerates to number of parameters in the worst case, but can be much smaller with structures on the data distribution.

A key observation is the minus quadratic term induced by the variance, which leads to self-modulating properties without explicit regularization. Technical results, including the general framework for training with perturbations, upper and lower bounds on the variance of dropout, and a new contraction argument for ReLU units with presence of denominators, are of independent interests.

2. Preliminaries

Notation: Throughout this paper, we use $A \circ B$ to denote entry-wise product of two matrices A and B with the same dimension. $A^{\circ 2}$ denotes entry-wise square of A, i.e., $A \circ A$. For vector $v \in \mathbb{R}^d$, we use $\mathrm{diag}\{v\}$ to denote the $d \times d$ diagonal matrix whose diagonal is v. Unless otherwise specified, σ is used for Rademacher random variables and g is used for Gaussian random variables. $[x]_+ = \max\{x, 0\}$ denotes the positive component of x. $\mathcal{P}$ denotes underlying distribution of data, and $\mathcal{P}_n$ denotes the empirical distribution. We slightly abuse the notation for expectation: $\mathbb{E}_{\mathcal{P}}$ denotes expectation under measure $\mathcal{P}$, while $\mathbb{E}_X$ means taking expectation with respect to the randomness of random variable X. We use $L(\cdot, \cdot)$ to denote the loss function. For any $c \in \mathbb{R}$ and function class $\mathcal{F}$, $c\mathcal{F} \triangleq \{c \cdot f : f \in \mathcal{F}\}$. For function class $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$ and offset functional $V(f, x) : \mathcal{F} \times \mathcal{X} \to \mathbb{R}$, we use $\hat{\mathcal{R}}_n(\mathcal{F}, V)$ to denote the empirical offset Rademacher complexity:

$$\hat{\mathcal{R}}_n(\mathcal{F}, V) \triangleq \mathop{\mathbb{E}}_{\sigma} \sup_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sigma_i f(x_i) - V(f, x_i) \right) \right\}.$$

where $\{\sigma_i\}_{i=1}^n$ are i.i.d. Rademacher random variables. The offset Rademacher complexity is defined as its expectation: $\mathcal{R}_n(\mathcal{F}, V) = \mathbb{E}\hat{\mathcal{R}}_n(\mathcal{F}, V)$. By replacing i.i.d. Rademacher random variables with i.i.d. Gaussian random variables, we also define $\hat{\mathcal{G}}_n(\mathcal{F}, V)$ and $\mathcal{G}_n(\mathcal{F}, V)$, the offset and true Gaussian complexities. The standard notion of offset Rademacher complexity in (Liang et al., 2015) can be seen as a special case where $V(f, x) = f(x)^2$.

2.1. Neural Networks and Dropout Training

Throughout this paper, the neural network models we are considering are fully-connected neural networks with ReLU activation. We define it as a model parametrized by a series of matrices $h = [W_1, W_2, \cdots, W_{L-1}, w_L]$ where $W_k \in \mathbb{R}^{d_{k-1} \times d_k}$ and $w_L \in \mathbb{R}^{d_{k-1}}$. The output of model is $h(x) = w_L^T \left[W_{L-1} [W_{L-2} \cdots [W_1 x]_+ \cdots]_+ \right]_+$. The dropout perturbation i.i.d. randomly discards entries of weight matrices in a neural network with probability q, while multiplying other entries by $\frac{1}{1-q}$.

3. Framework of Generalization Bounds for Training with Random Perturbations

In this section, we propose a statistical learning theory framework for the generalization properties of learning algorithms whose training procedure is accompanied with random perturbation. We will focus on generalization error of learning procedure. As opposed to (Wager et al., 2014), our theoretical results are distribution-free, since it doesn't depend upon any parametric assumptions on underlying distribution families.

In our analysis, the function h during training is perturbed by a random operator $\psi : \mathcal{H} \to \mathcal{H}$, and we consider the procedure of minimizing expected empirical risks:

$$\hat{h} = \mathop{\mathrm{argmin}}_{h \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{\psi} \left[L\left(\psi h(x_i), y_i \right) \right] \right\}, \tag{1}$$

where $\mathcal{H}$ is the hypothesis class and $L(\cdot, \cdot)$ is a loss function. We will be primarily focused on quadratic loss $L(z, y) = (z - y)^2$, while other loss functions will also be discussed.

In the case of dropout training, ψ uniformly and independently discards a fixed proportion q of matrix entries, while multiplying remaining entries with $\frac{1}{1-q}$. The dropout training algorithm can be conveniently viewed as stochastic gradient descent method for Optimization Problem (1), where we not only draw a batch of training examples randomly, but also generate samples from the perturbation operator ψ, in each iteration. Though dropout training is randomized, an algorithm that solves optimization Problem (1) needs not to be randomized. As the close form of expectation can be difficult, dropout training serves as an efficient algorithm for that regularized objective.

Throughout this paper, we refer to generalization error as the gap between population risk and expected empirical risk (under random perturbation), i.e.

$$\varepsilon_{gen} \triangleq \mathbb{E}_{\mathcal{P}} L(h(x), y) - \mathbb{E}_{\mathcal{P}_n} \mathbb{E}_{\psi} L(\psi h(x), y) \tag{2}$$

It is a common practice to use dropout during training and full network for testing, where the test error is compared against averaged training error with dropout. From a theoretical viewpoint, the asymmetry makes it impossible to bound the gap from both sides. As in (Liang et al., 2015), our method only considers the useful side of empirical process, which, fortunately, is the easy direction.

Theorem 1. *For $L(u, v) = \|u - v\|^2$ and unbiased random perturbation operator satisfying*

$$\forall h \in \mathcal{H}, x \in \mathcal{X}, \quad \mathop{\mathbb{E}}_{\psi} \psi h(x) = h(x).$$

Then we have the following upper bound:

$$\mathbb{E}\left(\varepsilon_{gen} \right) \leq 2\mathcal{R}_n \left(\mathcal{LH}, \frac{1}{4} V_\psi \right) + \mathcal{R}_n \left(V_\psi \mathcal{H}, \frac{1}{2} \mathrm{Id} \right). \tag{3}$$

where $\mathcal{LH} \triangleq \{(x, y) \rightarrowtail (h(x) - y)^2 : h \in \mathcal{H}\}$ is the class of squared loss functions; the offset operators are $V_\psi(h, x) = \mathrm{var}\left((\psi h)(x) \right)$, $\mathrm{Id}(h, x) = h(x)$, and the variance function class is $V_\psi \mathcal{H} = \{V_\psi(h, \cdot) : h \in \mathcal{H}\}$.

Theorem 1 controls the expected generalization error by sum of two offset Rademacher complexity terms:

- The main term $2\mathcal{R}_n \left(\mathcal{LH}, \frac{1}{4} V_\psi \right)$ has a minus offset compared to the Rademacher complexity of function class $\mathcal{LH}$, which comes from the variance of random perturbation. This characterizes the "randomization-as-regularization" effect. Even without norm-based assumptions, this minus quadratic term leads to self-modulating properties of this complexity term, and guarantees good generalization.
- The auxiliary term $\mathcal{R}_n \left(V_\psi \mathcal{H}, \frac{1}{2} \mathrm{Id} \right)$ comes from a more subtle reason: since the variance-based regularization is data-dependent, it is possible that $\sum_{i=1}^n V_\psi(h, x_i)$ gets small while the model complexity is still large. The regularization effect can be weakened in that case. Therefore, for characterizing regularization effects of variance, an empirical process control on V_ψ is unavoidable, which is measured by the standard offset Rademacher complexity.

We can also derive contraction results for the offset Rademacher complexities under Lipschitz function composition. This lemma will be repeatedly used in later theories.

Lemma 1. *Suppose $\phi : \mathbb{R} \rightarrowtail \mathbb{R}$ is l-Lipschitz. For any function class $\mathcal{F}$ and operator $V : \mathcal{F} \times \mathcal{X} \to \mathbb{R}$, we have:*

$$\hat{\mathcal{R}}_n(\phi(\mathcal{F}), V) \leq \hat{\mathcal{R}}_n(l \cdot \mathcal{F}, V), \tag{4}$$

where $\phi(\mathcal{F}) = \{x \rightarrowtail \phi(f(x)) : f \in \mathcal{F}\}$.

By applying Lemma 1 to the general result in Theorem 1, we obtain the following result for uniformly bounded class.

Theorem 2. *Under settings in Theorem 1, if we further assume $\sup_{h\in\mathcal{H}} \|h\|_\infty \leq a$, we have the following with probability $1-\delta$:*

$$\varepsilon_{gen} \leq 2\mathcal{R}_n(2a\mathcal{H}, \frac{1}{4}V_\psi) + \mathcal{R}_n(V_\psi\mathcal{H}, \frac{1}{2}\text{Id}) + O\left(\sqrt{\frac{1}{n}\log\frac{1}{\delta}}\right).$$

For loss functions other than quadratic loss, a minus quadratic term analogous to Theorem 1 and Theorem 2 can also be derived via local geometric structure of the loss function. We are particularly interested in classification problem, a general theoretical framework for which is stated in Theorem 3.

Theorem 3. *For the case of $y \in \{-1, 1\}$ and loss function $L(h(x), y) = g(yh(x))$ where $g(\cdot)$ is an l-Lipschitz, uniformly bounded convex function on some compact set. We have the following with probability $1-\delta$:*

$$\varepsilon_{gen} \leq 2\mathcal{R}_n(l\cdot\mathcal{H}, \frac{1}{4}\Delta_\psi) + \mathcal{R}_n(\Delta_\psi\mathcal{H}, \frac{1}{2}\text{Id}) + O\left(\sqrt{\frac{\log\frac{1}{\delta}}{n}}\right),$$

where $\Delta_\psi(h, z) = \mathbb{E}_\psi g(y \cdot \psi h(x)) - g(y \cdot h(x)) \geq 0$ for $z = (x, y)$, and $\Delta_\psi\mathcal{H} = \{\Delta_\psi(h, \cdot), h \in \mathcal{H}\}$.

The non-negative quantity $\Delta_\psi(h, z_i)$ characterizes the additional gain on generalization risk we can obtain by dropout with data point z_i. From a high level point of view, there is an interesting difference between the role of random perturbation in squared loss and the loss in Theorem 3: the random perturbation under squared loss induces a data-dependent regularization only based on the variance $\text{var}(\psi h(x))$, treating all data points with equal weights; the random perturbation under classification loss, on the other hand, assign unequal weights on the variances induced by each data point, by its closeness to decision boundary. This phenomenon can be clearly illustrated by the following proposition.

Proposition 1. *Let $L(h(x), y) = \log(1 + \exp(yh(x))$ be the logistic loss function. Perturbation operator ψ satisfies that: $\psi h(x_i)$ is symmetric around its expectation $h(x_i)$ for any h and x_i. Let $\text{var}_\psi\,(h(x_i)) = v_i(h)^2$, assuming that*

$$\text{var}\left(\psi h(x_i) \Big| |\psi h(x_i) - h(x_i)| \leq 2v_i(h)\right) \geq c v_i(h)^2,$$

we have:

$$\Delta_\psi(h, x_i) \geq \frac{c v_i(h)^2 \exp(h(x))}{(1 + e^{h(x_i)+4v_i(h)})(1 + e^{h(x_i)-4v_i(h)})} \tag{5}$$

Furthermore, taking the limit $v_i(h) \to 0$, we have $\Delta_\psi(h, x_i) = \Theta\left(\frac{v_i(h)^2 e^{h(x_i)}}{(1+e^{h(x_i)})^2}\right)$

Proposition 1 conveys some interesting information for dropout training under logistic loss: on the one hand, the additional offset term also essentially depends on the variance of model output induced by dropout, at least in low-variance regime; on the other hand, the offset term is putting a weight approximately $\frac{e^{h(x_i)}}{(1+e^{h(x_i)})^2}$ on the contribution from output variance at each data points x_i. A data point closer to decision boundary will induce a larger regularization term.

4. Dropout Training and Adaptive Regularization

According to Theorem 2, the generalization error of training under random perturbation is fundamentally controlled by the variance induced by random perturbation. In this section, we will study the variance of dropout in deep neural networks. First, a truthful dropout algorithm is proposed, to make the output unbiased with respect to perturbation operator. Variance induced by this algorithm is also studied, and connection to adaptive regularization and batch normalization methods are established.

4.1. Truthful Dropout and Variance Bounds

The unbiasedness assumption $\mathbb{E}_\psi \psi h(x) = h(x)$ in Theorem 2 is a fundamental and reasonable requirement, as we always want the predictor to be at least "truthful": the overall behavior of perturbed model used for training has to be close to the complete model for prediction. Due to the non-linear nature of activation function, however, the traditional dropout perturbation used in neural network is inherently biased. The common practice of multiplying remaining weights by $\frac{1}{1-q}$ can be viewed as a heuristic way of reducing this bias. But the bias still exists, since expectations are not preserved under nonlinear ReLU functions. We propose a truthful dropout algorithm (Algorithm 1) which yields unbiased output. The idea of this algorithm is very simple and the unbiasedness is obvious: we just compute the original feed-forwarding without dropout, and randomly reflect the value computed with dropout with the original one at each layer, so that the distribution is symmetric and unbiased. Later we will illustrate through experiments that this algorithm achieves good performance comparable to traditional dropout.

We then obtain the upper and lower bounds for variance induced by truthful dropout, using induction among layers.

Algorithm 1 Truthful dropout feed-forwarding

```
Input: ReLU network $h = \{W_1, W_2, \cdots, W_{L-1}, w_L\}$, input vector $x \in \mathbb{R}^{d_0}$, dropout probability $q$.
Output: Vectors in each layers $\left\{z^{(1)}, z^{(2)}, \cdots, z^{(L)}\right\}$.
  Let $\left\{u^{(1)}, u^{(2)}, \cdots, u^{(L)}\right\} = \mathsf{feedforward}(h, x)$.
  Let $z^{(0)} = x$.
  for $k \in \{1, 2, \cdots, L-1\}$ do
    Sample $\widetilde{W}_k = \frac{1}{1-q} \cdot \mathsf{dropout}(W_k)$.
    $\tilde{z}^{(k)} = \left[\widetilde{W}_k z^{(k-1)}\right]_+$.
    Sample $r \sim \mathcal{U}\{-1, 1\}^{d_k}$
    $z^{(k)} = u^{(k)} + r \circ (\tilde{z}^{(k)} - u^{(k)})$
  end for
  Let $z^{(L)} = \frac{1}{1-q} \cdot \mathsf{dropout}(w_L)^T z^{(L-1)}$.
```

Theorem 4. *Given a fixed input point $x_i \in \mathbb{R}^{d_1}$, let $v_i^{(k)} = \left(\mathrm{var}(z_i^{(k)})\right)_{i=1}^{d_k}$ be the variance vector induced by truthful dropout on k-th layer, we have the following for $k \geq 1$:*

$$v_i^{(k)} \geq \frac{1}{2} I_i^{(k)} W_k^{\circ 2} \left(\frac{1}{q} v_i^{(k-1)} + \frac{1-q}{q} \left(u_i^{(k-1)}\right)^{\circ 2}\right) \tag{6}$$

where $I_i^{(k)} = \mathrm{diag}\left\{1_{u_i^{(k)}(j) > 0}\right\}, v_i^{(0)} = 0, u_i^{(0)} = x_i$.

Theorem 5. *Given a fixed input point $x_i \in \mathbb{R}^{d_1}$, let $v_i^{(k)} = \left(\mathrm{var}(z_i^{(k)})\right)_{i=1}^{d_k}$ be the variance vector induced by truthful dropout on k-th layer, we have the following for $k \geq 1$:*

$$v_i^{(k)} \leq W_k^{\circ 2} \left(\frac{1}{q} v_i^{(k-1)} + \frac{1-q}{q} \left(u_i^{(k-1)}\right)^{\circ 2}\right) \tag{7}$$

The gap between upper and lower bounds lies in a constant factor increasing with layers, as well as the contribution by units zeroed-out by ReLU activation. In the next subsection, we will interpret the role of the estimated variances as data-dependent regularization.

4.2. Connections to Adaptive Regularization

For single-layer models, Wager et al., (Wager et al., 2013) has shown connection with dropout training with adaptive regularizers and subgradient methods for online convex optimization (Duchi et al., 2011). In the world of deep learning, neither of methods admit nice theoretical properties: dropout can be significantly different from ℓ_2 regularization (Helmbold & Long, 2017), while no theoretical results have been provided about why adaptive regularization works. However, the apparent difficulties in the analysis have never daunted deep learning practitioners: both methods are widely used for neural networks.

In this section, we will slightly deviate from the main theme of generalization error bounds, and establish a similar connection for arbitrarily deep ReLU networks. In the most general setup, though neither of the methods is known to imply good generalization error directly, this connection still provides important theoretical insights.

Following the convention of (Wager et al., 2013), we can explicitly write a data-dependent regularization method as online gradient method $\beta^+ = \arg\min_\theta \{-\eta \langle \nabla L(\beta; x_t), \theta \rangle + \|\beta - \theta\|_{A_t}\}$. The key issue is about the structure of matrix A_t that defines the data-dependent norm.

According to Theorem 4 and Theorem 5, the adaptive regularization is based on $\|\cdot\|_{A_t}$ norm, where

$$A_t \succeq \delta I + \sum_{i=1}^{t} \mathrm{diag}\left\{C_k \cdot \Pi_{l=k}^{L} (W_l^{\circ 2} I_i^{(l)}) \left(u_i^{(k-1)}\right)^{\circ 2}\right\}_{k=1}^{L}, \tag{8}$$

where C_k is a constant depending solely on k, and index $k = 1, 2, \cdots L$ denotes layers of the network.

The adaptive regularization $\|\cdot\|_{H_t}$ induced by AdaGrad, on the other hand, is based on sum of gradients, which can be easily calculated using back propagation:

$$H_t = \delta I + \left(\sum_{i=1}^{t} \mathrm{diag}\left\{\left(\Pi_{l=k}^{L} (W_l I_i^{(l)}) u_i^{(k-1)}\right)^{\circ 2}\right\}_{k=1}^{L}\right)^{\frac{1}{2}}, \tag{9}$$

Both A_t and H_t are written in the form of adaptive regularization, making them capable of adapting to non-isotropic geometric shape of data points' distribution. Similar to single layer case (Wager et al., 2013), AdaGrad takes square root of aggregated adaptive regularization matrix, while dropout doesn't. Moreover, dropout uses square of parameters along any connecting path in the network, while AdaGrad first calculates the gradient along path and then takes square. From this viewpoint, the adaptive regularization induced by dropout training can be seen as a data-dependent version of PathSGD (Neyshabur et al., 2015a), or a variant of the algorithm in (Neyshabur et al., 2015b).

Another widely-used technique in deep learning, batch normalization, is also closely related to adaptive regularization. In BN, data points in each layer are normalized to zero mean and identity covariance. We will briefly discuss its interaction with dropout.

On the one hand, if dropout is applied to a batch normalized network, where the point set $\{u_i^{(k)}\}_{i=1}^n$ has zero mean and identity covariance for each k, the data-dependent ℓ^2 norm induced by this set of points degenerates to the standard data-independent norm. On the other hand, batch normalization accompanied with standard ℓ_2 penalty will become a form of data-dependent regularization. Though different from dropout in many aspects, the heuristics about adaptivity to geometric shape are the same. This partially explains why BN can be seen as an alternative regularization method to dropout, and why two methods are not used at the same time (Ioffe & Szegedy, 2015).

5. Generalization Error Bounds

In this section, we apply the framework in Section 3 to truthful dropout methods in neural networks, and derive concrete generalization error guarantees based on the variance estimates in Section 4. We restrict our attention to neural networks with one hidden layer and ReLU activation, though the techniques are also potentially useful to deeper networks. Upper bounds for both terms in Theorem 2 will be derived under very mild assumptions. We will also discuss the practical implication of this bound and comparison with existing works.

Consider the ReLU network model class:

$$\mathcal{H} = \left\{ h : \mathbb{R}^d \to \mathbb{R}, h(x) \triangleq w^T [Wx]_+ \right\}, \qquad (10)$$

parametrized with $W \in \mathbb{R}^{p\times d}, w \in \mathbb{R}^d$. Note that we do not make any norm-based assumptions on the parameter space at all, which illustrate the self-modulating properties of variance induced by dropout. We need to assume output of this model on support of data distribution is bounded by a constant, i.e., $\sup_{h,x} |h(x)| \le a$, which is naturally true in deep learning practice.

The lower bound on dropout variance in Theorem 4 has some entries zeroed out by ReLU through $I_i(k)$, while the upper bound in Theorem 5 doesn't. This gap brings about additional difficulties to our analysis, and it's hard to obtain a more accurate estimate. However, as our primary goal is to understand the class of variance-induced data-dependent regularization, there is no need to stick to the original form of dropout. Indeed, dropout can be seen as a computationally efficient way of approximately achieving an idealized variance-induced regularizer. Therefore, we will take the offset operator $V_\psi(h,x) = \langle w^{\circ 2}, (Wx)^{\circ 2} + W^{\circ 2} x^{\circ 2}\rangle$. It is also easy to construct a random perturbation operator ψ to satisfy this, by artificially injecting Gaussian noises on the units zeroed out by dropout. Though $V_\psi(h,x)$ is not the actual variance, we will still be assuming $0 \le V_\psi(h,x) \le a^2, \forall h, x$, as is naturally satisfied in deep learning practice.

By making above assumptions and simplifications, we have reached a clear setup where concrete bounds with reasonable practical implication can be shown. According to Theorem 2, it remains to derive upper bounds on $\mathcal{R}_n(2a\mathcal{H}, \frac{1}{4}V_\psi)$ and $\mathcal{R}_n(V_\psi \mathcal{H}, \frac{1}{2}\mathrm{Id})$. In the following few subsections, we will prove these bounds for the one-hidden-layer model, and discuss possible approaches towards deeper networks.

We need the following technical lemma that relates offset versions of Rademacher and Gaussian complexities:

Lemma 2. *For any function class $\mathcal{F}$ and operator V : $\mathcal{F} \times \mathcal{X} \to \mathbb{R}$, we have:*

$$\hat{\mathcal{R}}_n(\mathcal{F}, V) \le \hat{\mathcal{G}}_n\left(\sqrt{\frac{\pi}{2}}\mathcal{F}, V\right). \qquad (11)$$

5.1. Bounding the Offset Complexity for $\mathcal{H}$

First of all, we have $\mathcal{R}_n(2a\mathcal{H}, \frac{1}{4}V_\psi) \le \mathcal{G}_n(a\sqrt{2\pi}\mathcal{H}, \frac{1}{4}V_\psi)$ by Lemma 2. So we only need to derive upper bounds on $\hat{\mathcal{G}}_n(\mathcal{H}, c_0 V_\psi)$ for absolute constant $c_0 = \frac{1}{8a}\sqrt{\frac{1}{2\pi}}$. Our proof strategy is to directly solve for the value of w that achieves supremum for the second layer, and then use contraction-type arguments to overcome the non-linearity in the first layer. Unlike Lemma 1, this step requires the powerful Gaussian Comparison Theorems, so the offset Gaussian complexity is used instead of Rademacher counterpart.

The last layer of the network is a linear model, and the supremum with respect to w in the form of offset Gaussian complexity can be directly solved out as following:

Lemma 3. *For the neural network class $\mathcal{H}$ and offset term $V_\psi(h,x)$ defined above, we have:*

$$\hat{\mathcal{G}}_n(\mathcal{H}, c_0 V_\psi) \le \frac{p}{c_0}\mathbb{E}\sup_{\beta\in\mathbb{R}^d} \frac{\left(\frac{1}{n}\sum_{i=1}^n g_i[\beta^T x_i]_+\right)^2}{\frac{1}{n}\sum_{i=1}^n \left((\beta^T x_i)^2 + \left(\beta^{\circ 2}\right)^T x_i^{\circ 2}\right)} \qquad (12)$$

It remains to bound RHS of Eq. (12) from above, which is the complexity control of first layer with nonlinear activation. The key difficulties come from the nonlinear ReLU unit in neural networks. Existing works on norm-based generalization guarantees in neural networks, such as (Neyshabur et al., 2015c) use Ledoux-Talagrand Contraction Lemma to deal with that. However, the denominator in Eq. (12) is not adaptable to standard contraction lemmas. To tackle this issue, we turn to use Gaussian Comparison Theorem, which can be found, for example, in (Ledoux & Talagrand, 2013). Actually, the contraction doesn't hold for arbitrary Lipschitz activation function. Fortunately, for ReLU units an inequality can be directly proven, leading to the following contraction result, for which the proof is postponed to the Appendix.

Lemma 4. *For any function class $\mathcal{F}$ and non-negative function $S(f, \{x_i\}_{i=1}^n)$, we have:*

$$\begin{aligned}&\mathbb{E}\sup_{f\in\mathcal{F}}\left\{\frac{\left(\frac{1}{n}\sum_{i=1}^n g_i[f(x_i)]_+\right)^2}{\frac{1}{n}\sum_{i=1}^n f(x_i)^2 + S(f,\{x_i\}_{i=1}^n)}\right\}\\ \le&4\mathbb{E}\sup_{f\in\mathcal{F}}\left\{\frac{\left(\frac{1}{n}\sum_{i=1}^n g_i f(x_i)\right)^2}{\frac{1}{n}\sum_{i=1}^n f(x_i)^2 + S(f,\{x_i\}_{i=1}^n)}\right\} + \frac{6}{n}\end{aligned}$$

It is worth noticing that the lemma holds for arbitrary function class $\mathcal{H}$ and non-negative function S. This makes

Lemma 4 potentially useful for dropout training in deeper ReLU networks. In the case of RHS of Eq. (12), we use the function class $\mathcal{F} = \{x \rightarrowtail \beta^T x : \beta \in \mathbb{R}^d\}$ and $S(f, \{x_i\}_{i=1}^n) = \frac{1}{n}\sum_{i=1}^n \left(\beta^{\circ 2}\right)^T x_i^{\circ 2}$.

Putting them together, we can control the first term in Theorem 2 for our model:

Theorem 6. *For neural network model $\mathcal{H}$ and offset term V_ψ defined above, for any constant $c_0 > 0$, we have:*

$$\hat{\mathcal{R}}_n(\mathcal{H}, c_0 V_\psi) \leq O\left(\frac{p}{n}\mathrm{tr}\left(\left(\bar{S}_n + \bar{D}_n\right)^{-1}\bar{S}_n\right)\right) \quad (13)$$

where $\bar{D}_n = \frac{1}{n}\mathrm{diag}\left(\sum_{i=1}^n x_i^{(2)}\right)$, $\bar{S}_n = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$.

Though still depending on the dimension of hidden layer, Theorem 6 achieves $O(1/n)$ fast rate with neither localization conditions nor norm-based constraints. We will discuss its implications in later subsections.

5.2. Bounding the Offset Complexity for $V_\psi \mathcal{H}$

We now proceed to derive upper bounds on the second term $\mathcal{R}(V_\psi\mathcal{H}, \frac{1}{2}\mathrm{Id})$. In our setup, $V_\psi(h, x) = \left(w^{\circ 2}\right)^T (Wx)^{\circ 2} + \left(w^{\circ 2}\right)^T W^{\circ 2} x^{\circ 2}$ has two terms. Let function classes $\mathcal{V}_1 = \{x \rightarrowtail \left(w^{\circ 2}\right)^T (Wx)^{\circ 2}\}$ and $\mathcal{V}_2 = \{x \rightarrowtail \left(w^{\circ 2}\right)^T W^{\circ 2} x^{\circ 2}\}$ be parametrized by $w \in \mathbb{R}^d$ and $W \in \mathbb{R}^{p\times d}$. We slightly abuse the notation by still using $V_\psi(h,x)$ to denote the original offset terms, though the actual function $h \in \mathcal{V}_i$ is different from those in $\mathcal{H}$. Our strategy is to bound the two terms $\hat{\mathcal{R}}_n(\mathcal{V}_1, \frac{1}{4}V_\psi)$ and $\hat{\mathcal{R}}_n(\mathcal{V}_2, \frac{1}{4}V_\psi)$ individually, which directly leads to the upper bound on $\hat{\mathcal{R}}_n(V_\psi\mathcal{H}, \frac{1}{2}\mathrm{Id})$. To estimate $\hat{\mathcal{R}}_n(\mathcal{V}_i, \frac{1}{4}V_\psi)$, we notice that these Rademacher processes are actually coordinate-wise separable. We can make a very crude estimate by using the uniform upper bound on the sum as the Lipscthiz constant for quadratic functions at each coordinate, and use Lemma 1 for the contraction arguments. Despite the looseness, our bounds for $\mathcal{R}(V_\psi\mathcal{H}, \frac{1}{2}\mathrm{Id})$ is usually no larger than Theorem 6, as shown in Theorem 7. The proof details are postponed to the Appendix.

Theorem 7. *For function class $\mathcal{H}$ and offset term V defined above, we have:*

$$\hat{\mathcal{R}}_n(V_\psi\mathcal{H}, \frac{1}{2}\mathrm{Id}) \leq O\left(\frac{p}{n}\mathrm{tr}\left(\left(\bar{S}_n + \bar{D}_n\right)^{-1}\bar{S}_n\right) + \frac{d}{n}\right) \quad (14)$$

where $\bar{S}_n$ and $\bar{D}_n$ are defined in the same way as Theorem 6.

5.3. Discussion about the Bounds

Putting everything together, we can derive the main generalization bound for two-layer neural networks:

Theorem 8. *Under the boundedness assumption and definition of idealized V_ψ as above, we have the following with probability $1-\delta$:*

$$\varepsilon_{gen} \leq O\left(\frac{p}{n}\mathrm{tr}\left(\mathbb{E}\left(\bar{S}_n + \bar{D}_n\right)^{-1}\bar{S}_n\right) + \frac{d}{n} + \sqrt{\frac{\log 1/\delta}{n}}\right). \quad (15)$$

Note that $\mathrm{tr}\left(\left(\bar{S}_n + \bar{D}_n\right)^{-1}\bar{S}_n\right) \leq \mathrm{tr}(I) = d$ in the worst case. So the generalization bound in Theorem 8 is at most $O(\frac{pd}{n} + \sqrt{\frac{\log 1/\delta}{n}})$, which has a linear dependence on the number of parameters, and roughly corresponds to bounds based on VC dimension (Harvey et al., 2017). Even in the worst case, however, the regularization effect of dropout still has important impact on the generalization error. More importantly, in many interesting scenarios, the bound can be adaptive to geometry of data and give much better results. In the following we will discuss these advantages.

Fast $O(1/n)$ rate: Classical empirical process theories without the offset term usually yield bounds in the form of $O(\sqrt{\frac{\text{Complexity}}{n}})$ for parametric classes. (For example, the complexity term can be VC dimension, metric entropy, or chaining-based estimates, etc.). This is fundamentally due to the Massart's finite class lemma for Rademacher complexities which gives $O(1/\sqrt{n})$ rate. Localization techniques, such as (Bartlett et al., 2005), are able to control the generalization error by considering a subclass $\mathcal{H} \cap \mathcal{B}_2(h^*, \delta_n)$ around h^* and achieve fast rate of convergence. (Though we still get $O(1/\sqrt{n})$ concentration term, this term involves no complexity measures.) Deep neural network model, as is well known, has extremely non-convex objective function, which makes it particularly hard even to talk about proximity to global optimum or star-shaped structure. However, if we are comparing test loss with expected training loss under the perturbation, such an $O(1/n)$ rate can be achieved from the offset term induced by the variance. This illustrates a similar type of self-modulating properties as in (Liang et al., 2015), but makes no assumptions about h^*. This initiates an attempt for fast rates in non-convex neural network models.

Adaptivity to Geometry of Data Points: In many scenarios, the term $\mathrm{tr}\left(\left(\bar{S}_n + \bar{D}_n\right)^{-1}\bar{S}_n\right)$ can be much smaller than the worst case upper bound. This is an important feature of data-dependent regularization: the induced generalization error is adaptive to the geometry of data points. This is in contrast with the ordinary ℓ_2 regularization. To illustrate the adaptivity effect, we give an upper bound for this quantity in the special case where all data points are lying in a low-dimensional subspace.

Proposition 2. *If $x_1, x_2, \cdots, x_n \in U \subseteq \mathbb{R}^d$ with $\dim(U) = r$. Assume that $\bar{S}_n + \bar{D}_n$ is invertible, we have:*

$$\mathrm{tr}\left(\left(\bar{S}_n + \bar{D}_n\right)^{-1}\bar{S}_n\right) \leq r, \quad (16)$$

Table 1. Comparison of test error using different networks on MNIST and CIFAR-10

Data set	Networks	Truthful Dropout	Traditional Dropout	No Dropout
MNIST	Fully-Connected	1.60	**1.53**	1.78
MNIST	Shallow Convolutional	0.99	**0.81**	1.16
CIFAR-10	Deep Convolutional	**8.20**	8.39	9.86

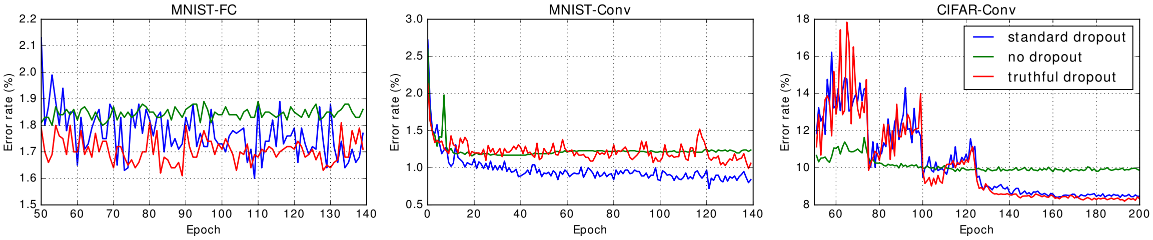

Figure 1. Comparison of test error along optimization trajectory

and therefore, with probability $1 - \delta$ *we have:*

$$\varepsilon_{gen} \leq O\left(\frac{rp + d}{n} + \sqrt{\frac{\log 1/\delta}{n}}\right). \tag{17}$$

If the data matrix is approximately low-rank, this geometric quantity will still be controlled at a low scale, as $\mathrm{tr}((\bar{S}_n + \bar{D}_n)^{-1}\bar{S}_n)$ is continuous. Other geometric structures also lead to small value of this quantity potentially.

From a high-level point of view, Theorem 8 illustrates two aspects of the effect of dropout: on the one hand, the ”altitude training” phenomenon really works in nonlinear neural networks in an assumption-free way, which achieves fast rate by the self-modulating term; on the other hand, the data-dependent regularization led by its variance helps adaptivity to the geometry of underlying distribution.

6. Experiments

In this section, we conduct experiments to verify the effectiveness of Algorithm 1, the truthful dropout, demonstrating the comparable generalization improvement in terms of classification error when comparing with standard Dropout.

We use MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky & Hinton, 2009) datasets to test our algorithm, with both convolutional and fully-connected neural networks. The details about experimental setup are postponed to the Appendix.

The classification error on test set is shown in the Table 1. We also plot the curve for classification error during optimization in the Figure 1. We can see that the error gap between traditional dropout and truthful dropout is relatively smaller than that between traditional dropout and no dropout (0.07% v.s 0.25% and 0.18% v.s 0.35%) on MNIST data set. On CIFAR-10 data set, truthful dropout even outperforms traditional dropout and gains 0.19% more accuracy.

7. Conclusion

The learning procedure with random perturbation during training is comprehensively studied in this paper, with dropout training as a prominent example. A distribution-free theory is first proposed, to characterize the role of perturbation's variance, from a statistical learning theory perspective. In particular, the generalization error is upper bounded by sum of two offset Rademacher complexity terms. The first one appends minus quadratic terms depending on variance of perturbation compared to the standard Rademacher complexity, which illustrate the self-modulating properties of data-dependent regularization led by this variance. The second term is the offset Rademacher complexity of variance, characterizing how the change in variance weaken the regularization effect.

For dropout training in neural networks, we first propose a truthful dropout algorithm that has unbiased output, the variance of which is analyzed with upper and lower bounds. Our bound has a clear relationship to adaptive regularization methods such as AdaGrad, and the estimated variance also explains the relationship between dropout training and batch normalization. Using our framework, we prove an upper bound for the generalization error for one-hidden-layer ReLU neural networks with truthful dropout. This bound achieves $O(1/n)$ fast rate and is adaptive to geometric structure of input data points. It is the first one that captures the ”altitude training” and data-dependent regularization effect of dropout in non-linear neural network models. An important future work is to extend our analysis to deeper neural networks and get concrete generalization bounds.

Acknowledgment

This work was partially supported by National Basic Research Program of China (973 Program) (grant no. 2015CB352502), NSFC (61573026) and Center for Data Science, Beijing Institute of Big Data Research in Peking University.

References

Bartlett, P. L., Bousquet, O., Mendelson, S., et al. Local rademacher complexities. *The Annals of Statistics*, 33(4): 1497–1537, 2005.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

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

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension bounds for piecewise linear neural networks. *arXiv preprint arXiv:1703.02930*, 2017.

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

Helmbold, D. P. and Long, P. M. Surprising properties of dropout in deep networks. In *Conference on Learning Theory*, pp. 1123–1146, 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.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pp. 646–661. Springer, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ledoux, M. and Talagrand, M. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

Liang, T., Rakhlin, A., and Sridharan, K. Learning with square loss: Localization through offset rademacher complexity. In *Proceedings of The 28th Conference on Learning Theory*, pp. 1260–1285, 2015.

Ma, X., Gao, Y., Hu, Z., Yu, Y., Deng, Y., and Hovy, E. Dropout with expectation-linear regularization. *arXiv preprint arXiv:1609.08017*, 2016.

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., Salakhutdinov, R., and Srebro, N. Data-dependent path normalization in neural networks. *arXiv preprint arXiv:1511.06747*, 2015b.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. *arXiv preprint arXiv:1503.00036*, 2015c.

Rippel, O., Gelbart, M., and Adams, R. Learning ordered representations with nested dropout. In *International Conference on Machine Learning*, pp. 1746–1754, 2014.

Van Erven, T. and Kotl, W. Follow the leader with dropout perturbations. In *COLT*, pp. 949–974, 2014.

Wager, S., Wang, S., and Liang, P. S. Dropout training as adaptive regularization. In *Advances in neural information processing systems*, pp. 351–359, 2013.

Wager, S., Fithian, W., Wang, S., and Liang, P. S. Altitude training: Strong bounds for single-layer dropout. In *Advances in Neural Information Processing Systems*, pp. 100–108, 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.