Capacity Control of ReLU Neural Networks by Basis-Path Norm

Shuxin Zheng,[1] **Qi Meng,**[2] **Huishuai Zhang,**[2] **Wei Chen,**[2] **Nenghai Yu,**[1] **Tie-Yan Liu**[2]
[1]University of Science and Technology of China
[2]Microsoft Research Asia
zhengsx@mail.ustc.edu.cn, {meq, huzhang, wche}@microsoft.com, ynh@ustc.edu.cn, Tie-Yan.Liu@microsoft.com

Abstract

Recently, path norm was proposed as a new capacity measure for neural networks with Rectified Linear Unit (ReLU) activation function, which takes the rescaling-invariant property of ReLU into account. It has been shown that the generalization error bound in terms of the path norm explains the empirical generalization behaviors of the ReLU neural networks better than that of other capacity measures. Moreover, optimization algorithms which take path norm as the regularization term to the loss function, like Path-SGD, have been shown to achieve better generalization performance. However, the path norm counts the values of all paths, and hence the capacity measure based on path norm could be improperly influenced by the dependency among different paths. It is also known that each path of a ReLU network can be represented by a small group of linearly independent basis paths with multiplication and division operation, which indicates that the generalization behavior of the network only depends on only a few basis paths. Motivated by this, we propose a new norm *Basis-path Norm* based on a group of linearly independent paths to measure the capacity of neural networks more accurately. We establish a generalization error bound based on this basis path norm, and show it explains the generalization behaviors of ReLU networks more accurately than previous capacity measures via extensive experiments. In addition, we develop optimization algorithms which minimize the empirical risk regularized by the basis-path norm. Our experiments on benchmark datasets demonstrate that the proposed regularization method achieves clearly better performance on the test set than the previous regularization approaches.

Introduction

Deep neural networks have pushed the frontiers of a wide variety of AI tasks in recent years such as speech recognition (Xiong et al. 2016; Chan et al. 2016), computer vision (Ioffe and Szegedy 2015; Ren et al. 2015) and neural language processing (Bahdanau, Cho, and Bengio 2014; Gehring et al. 2017), etc. More surprisingly, deep neural networks generalize well, even when the number of parameters is significantly larger than the amount of training data (Zhang et al. 2017). To explain the generalization ability of neural networks, researchers commonly used different norms of network parameters to measure the capacity (Bartlett, Foster, and Telgarsky 2017; Neyshabur, Tomioka, and Srebro 2015; 2016).

Among different types of deep neural networks, ReLU networks (i.e., neural networks with ReLU activations (Glorot, Bordes, and Bengio 2011)) have demonstrated their outstanding performances in many fields such as image classification (He et al. 2016; Huang et al. 2017), information system (Cheng et al. 2016; He et al. 2017), and text understanding (Vaswani et al. 2017) etc. It is well known that ReLU neural networks are positively scale invariant (Neyshabur, Salakhutdinov, and Srebro 2015; Neyshabur et al. 2016). That is, for a hidden node with ReLU activation, if all of its incoming weights are multiplied by a positive constant c and its outgoing weights are divided by the same constant, the neural network with the new weights will generate exactly the same output as the old one for any arbitrary input. (Neyshabur, Salakhutdinov, and Srebro 2015) considered the product of weights along all paths from the input to output units as path norm which is invariant to the rescaling of weights, and proposed Path-SGD which takes path norm as the regularization term to the loss function.

In fact, each path in a ReLU network can be represented by a small group of generalized linearly independent paths (we call them *basis-path* in the sequels) with multiplication and division operation as shown in Figure 1. Thus, there is dependency among different paths. The smaller the percentage of basis paths, the higher the dependency. As the network is determined only by the basis paths, the generalization property of the network should depend only on the basis paths, as well as the relevant regularization methods. In addition, Path-SGD controls the capacity by solving argmin of the regularized loss function, the solution of the argmin problem is approximate because dependency among different values of all paths is not considered in the network. This motivates us to establish a capacity bound based on only the basis paths instead of all the paths. This is in contrast to the generalization bound based on the path norm which counts the values of all the paths and does not consider the dependency among different paths. To tackle these problems, we define a new norm based on the values of the basis paths called *Basis-path Norm*. In previous work, (Meng et al. 2018) constructed the basis paths by *skeleton method* and proved that the values of all other paths can be calculated using the values of basis paths by multiplication and division operations. In this work, we take one step further and categorize the basis paths into *positive* and *negative* basis paths according to the sign of their coefficients in the calculations of non-basis paths.

In order to control generalization error, we need to keep the hypothesis space being small. As we know, loss function can be computed by paths, hence we keep the values of all paths being small. To keep small values of non-basis paths represented by positive and negative basis paths, we control the positive basis paths not being too large while the negative basis paths not being too small. In addition, to keep small values of basis paths, we control the negative basis paths not being too large as well. With this consideration, we define the new Basis-path norm. We prove a generalization error bound for ReLU networks in terms of the basis-path norm. We then study the relationship between this basis-path norm bound and the empirical generalization gap - the absolute difference between test error and training error. The experiments included ReLU networks with different depths, widths, and levels of randomness to the label. For comparison purpose, we also compute the generalization error bounds induced by other capacity measures for neural networks proposed in the literature. Our experiments show that the generalization bound based on basis-path norm is much more consistent with the empirical generalization gap than those based on other norms. In particular, when the network size is small, the ordinary path norm bound fit empirical generalization gap well. However, when the width and depth increases, the percentage of non-basis paths increases, and the dependency among paths increases and we observe that the path norm bound degenerates in reflecting the empirical generalization gap. In contrast, our basis-path norm bound fits the empirical generalization gap consistently as the network size changes. This validates the efficacy of BP norm as a capacity measure.

Finally, we propose a novel regularization method, called Basis-path regularization (BP regularization), in which we penalize the loss function by the BP norm. Empirically, we first conduct experiments on recommendation system of MovieLens-1M dataset to compare the multi-layer perceptron (MLP) model's generalization with BP regularization and baseline norm-based regularization, then we verify the effectiveness of BP regularization on image classification task with ResNet and PlainNet on CIFAR-10 dataset. The results of all experiments show that, with our method, optimization algorithms (i.e., SGD, Adam, Quotient SGD) can attain better test accuracy than with other regularization methods.

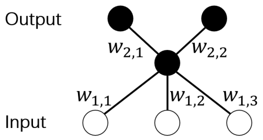

Figure 1: A toy neural network example. The network has 6 paths $p_{i,j}$, where $i \in \{1,2,3\}$ and $j \in \{1,2\}$, the values of paths $v_{p_{i,j}} = w_{1,i}w_{2,j}$, We can see the dependency among the paths, i.e., $v_{p_{2,2}} = \frac{v_{p_{1,2}} \cdot v_{p_{2,1}}}{v_{p_{1,1}}}$ and $v_{p_{3,2}} = \frac{v_{p_{3,1}} \cdot v_{p_{1,2}}}{v_{p_{1,1}}}$. In this group of basis paths, $p_{1,1}$ is the negative Basis-path, $p_{1,2}$,$p_{2,1}$ and $p_{3,1}$ are the positive basis paths.

Related Work

Generalization of deep neural networks has attracted a great deal of attention in the community (Zhang et al. 2017; Neyshabur et al. 2017; Kawaguchi, Kaelbling, and Bengio 2017). Norm and margin-based measures have been widely studied, and commonly used in neural network optimization with capacity control (Bartlett and Mendelson 2002; Evgeniou, Pontil, and Poggio 2000; Neyshabur, Tomioka, and Srebro 2016). For example, in (Bartlett, Foster, and Telgarsky 2017), the authors proposed a margin-based generalization bound for networks that scale with their margin-normalized spectral complexity. An analysis of generalization bounds based on PAC-Bayes was proposed in (Dziugaite and Roy 2017).

Among these measures, the generalization bound based on path norm is tighter theoretically (Neyshabur, Tomioka, and Srebro 2016). Empirically, path norm has been showed to be more accurate to describe the tendency of generalization error (Neyshabur et al. 2017). Thus, we are interested in the capacity measure which is related to the path norm. In (Neyshabur, Tomioka, and Srebro 2016), the authors first proposed group norm and path norm. The results show that the path norm is equivalent to a kind of group norm. In (Neyshabur, Salakhutdinov, and Srebro 2015; Neyshabur et al. 2016), the authors proposed to use path norm as a regularization term for ReLU multi-layers perceptron (MLP) network and recurrent network and designed Path-SGD algorithm. In (Neyshabur et al. 2017), the authors empirically compared different kinds of capacity measures including path norm for deep neural network generalization. However, none of those norms considered the dependency among paths in the networks.

Preliminaries

In this section, we introduce ReLU neural networks and generalization error. First of all, we briefly introduce the structure of rectifier neural network models. Suppose $f_w : \mathcal{X} \to \mathcal{Y}$ is a L-layer neural network with weight $w \in \mathcal{W}$, where input space $\mathcal{X} \subset \mathbb{R}^d$ and output space $\mathcal{Y} = \mathbb{R}^K$. In the l-th layer ($l = 0, ..., L$), there are h_l nodes. We denote the nodes and their values as $\{O^l, o^l\}$. It is clear that, $h_0 = d, h_L = K$. The layer mapping is given as, $o^l = \sigma(w_l^T o^{l-1})$, where w_l is the adjacency matrix in the l-layer, and the rectifier activation function $\sigma(\cdot) = max(\cdot, 0)$ is applied element-wisely. We can also calculate the k-th output by paths, i.e.,

$$N_w^k(x) = \sum_{(i_0, \cdots, i_L = k)} \prod_{l=1}^{L} w_l(i_{l-1}, i_l) \cdot \prod_{l=1}^{L-1} \mathbb{I}(o_{i_l}^l(w,x) > 0) \cdot x_{p_0} \tag{1}$$

where $(i_0, \cdots, i_L)$ is the path starting from input feature node $O_{i_0}^0$ to output node $O_{i_L}^L$ via hidden nodes

$O^1_{i_1}, ..., O^{L-1}_{i_{L-1}}$, and $w_l(i_{l-1}, i_l)$ is the weight of the edge connecting nodes $O^{l-1}_{i_{l-1}}$ and $O^l_{i_l}$. [1]

We denote $p_{(i_0,\cdots,i_L)} = \prod_{l=1}^L w_l(i_{l-1}, i_l)$ and $a_{(i_0,\cdots,i_L)} = \prod_{l=1}^{L-1} \mathbb{I}(o^l_{i_l}(w,x) > 0)$. The output can be represented using paths as

$$N^k_{p,a}(x) = \sum_{(i_0,\cdots,i_L)} p_{(i_0,\cdots,i_L)} \cdot a_{(i_0,\cdots,i_L)} \cdot x_{i_0}.$$

For ease of reference, we omit the explicit index $(i_0, \cdots, i_L)$ and use i be the index of path. We use $p = (p_1, p_2, \cdots, p_M)$ where $M = \prod_{l=0}^L h_l$ to denote the path vector. The path norm used in Path-SGD (Neyshabur, Salakhutdinov, and Srebro 2015) is defined as $\Omega(p) = \left(\sum_{i=1}^M p_i^2\right)^{1/2}$.

Given the training set $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ i.i.d sampled from the underlying distribution $\mathbb{P}$, machine learning algorithms learn a model f from the hypothesis space $\mathcal{F}$ by minimizing the empirical loss function $l(f(x), y)$. The uniform generalization error of empirical risk minimization in hypothesis space $\mathcal{F}$ is defined as

$$\epsilon_{gen}(\mathcal{F}) = \sup_{f\in\mathcal{F}} |\frac{1}{n}\sum_{i=1}^n l(f(x_i), y_i) - \mathbb{E}_{(x,y)\sim\mathbb{P}} l(f(x), y)|.$$

Generalization error ϵ_{gen} measures how well a model f learned from the training data S can fit an unknown test sample $(x, y) \sim \mathbb{P}$.

Empirically, we consider the empirical generalization error which is defined as the difference of empirical loss between the training set and test set at the trained model f.

Basis-path Norm

In this section, we define the Basis-path Norm (abbreviated as BP norm) on the networks. Using the BP norm, we define a capacity measure which is called BP-measure and we prove that the generalization error can be upper bounded using this measure.

The Definition of Basis-path Norm

First, as shown in (Meng et al. 2018), the authors constructed a group of basis paths by skeleton method[2]. It means that the value of non-basis paths can be calculated using the values of basis paths. In the calculation of non-basis paths' values, some basis paths always have positive exponents and hence appear in the numerator, others have negative exponents and hence appear in the denominator. We use $\tilde{p}$ to denote a non-basis path and $p_1, \cdots, p_r$ to denote basis paths. We have the following proposition.

Proposition 1 *For any non-basis path $\tilde{p}$, $\tilde{p} = \prod_{i=1}^m p_i^{\alpha_i} \prod_{j=m+1}^r p_j^{\alpha_j}$, where $\alpha_i \leq 0, \alpha_j \geq 0$.*

Limited by the space, we put the detailed proof in the appendices.

The proposition shows that basis paths $p_1, \cdots, p_m$ always have negative exponent in the calculation, while $p_{m+1}, \cdots, p^r$ always have positive exponent. We call the basis path with negative exponent α_i *Negative Basis Path* and denote the negative basis path vector as $p^- = (p_1, \cdots, p_m)$. We call the basis path with positive exponent α_j *Positive Basis Path* and denote it as $p^+ = (p_{m+1}, \cdots, p_r)$.

In order to control generalization error, we need to keep the hypothesis space being small. Thus we want all the paths to have small values. For non-basis path represented by p_i and p_j, we control p_i not being too small because α_i is negative, and p_j not being too large because α_j is positive. We control p_i not being too large as well to keep small values of basis paths. We define the following basis-path norm $\phi(\cdot)$ as follows.

Definition 1 *The basis norm on the ReLU networks is*

$$\phi(p) = \sup\{|\log|p_1||, \cdots, |\log|p_m||, |p_{m+1}|, \cdots, |p_r|\}. \tag{2}$$

We next provide the property of $\phi(p)$.

Theorem 1 *$\phi(p)$ is a norm in the vector space where p^+ is a vector in Euclidean space and p^- is a vector in a generalized linear space under the generalized addition and generalized scalar multiplication operations: $p^- \oplus (p')^- = [p_1 \cdot p'_1, \cdots, p_m \cdot p'_m]$ and $c \odot p^- = [sgn(p_1) \cdot |p_1|^c, \cdots, sgn(p_m) \cdot |p_m|^c]$ for $p^-, (p')^- \in \mathbb{R}^m$ and $c \in \mathbb{R}$.*

Proof: The definition of $\phi(p)$ is equivalent to

$$\phi(p) = \sup\{\phi_\infty(p^+), \phi_\infty(p^-)\}. \tag{3}$$

where $\phi_\infty(p^-) = \sup_i \{|\log|p_i||, i = 1, \cdots, m\}$ and $\phi_\infty(p^+) = \sup_j \{|p_j|, j = m+1, \cdots, r\}$. Obviously, $\phi_\infty(p^+)$ is the ℓ_∞ norm in Euclidean space. Thus, it only needs to prove $\phi_\infty(p^-)$ is a kind of norm. Next, we prove that $\phi_\infty(p^-)$ is a norm in the generalized linear space.

In the generalized linear space, the zero vector is I, where I denotes a vector with all elements being equal to 1. Based on the generalized linear operators, we verify the properties including positive definite, absolutely homogeneous and the triangle inequality of $\|p^-\|_\infty$ as follows:

(1) (Positive definite) $\phi_\infty(p^-) \geq 0$ and $\phi_\infty(p^-) = 0$ when $p^- = I$.

(2) (Absolutely homogeneous) For arbitrary $c \in \mathbb{R}$, we have

$$\phi_\infty(c \cdot p^-) = \sup_i \left\{|\log|p_i^-|^c|, i = 1, \cdots, m\right\} = |c| \cdot \phi_\infty(p^-).$$

(3) (Triangle inequality)

$$\begin{aligned}\phi_\infty(p^- \oplus (p')^-) =& \sup_i \{|\log|p_i p'_i||, i = 1, \cdots, m\} \\ \leq& \sup_i \{|\log|p_i||, i = 1, \cdots, m\} \\ &+ \sup_i \{|\log|p'_i||, i = 1, \cdots, m\} \\ =& \phi_\infty(p^-) + \phi_\infty((p')^-).\end{aligned}$$

Considered that $\phi_\infty(p^+)$ and $\phi_\infty(p^-)$ are both norms, taking supreme of them is still a norm. Thus $\phi(p)$ satisfies the definition of norm. □

[1] The paths across the bias node can also be described in the same way. For simplicity, we omit the bias term.

[2] Please note that different basis vectors in a vector space are equivalent and can be converted to each other.

Generalization Error Bound by Basis-path Norm

We want to use the basis-path norm to define a capacity measure to get the upper bound for the generalization error. Suppose the binary classifier is given as $g(x) = v^T f(x)$, where v represents the linear operator on the output of the deep network with input vector $x \in \mathbb{R}^d$. We consider the following hypothesis space which is composed of linear operator v, and L-layered fully connected neural networks with width H and input dimension d:

$$\mathcal{G}_{\gamma,v}^{d,H,L} =$$
$$\{g = v \circ f : L \geq 2, h_0 = d, h_1 = \cdots = h_{L-1} = H, \phi(p) \leq \gamma\}.$$

Theorem 2 *Given the training set $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ with $x_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$, and the hypothesis space $\mathcal{G}_{\gamma,v}^{d,H,L}$ which contains MLPs with depth $L \geq 2$, width H and $\phi(p)$, for arbitrary $z > 0$, for every $\delta > 0$, with probability at least $1 - \delta$, for every hypothesis $g \in \mathcal{G}_{\gamma,v}^{d,H,L}$, the generalization error can be upper bounded as*

$$\epsilon_{gen}(\mathcal{G}_{\gamma,v}^{d,H,L}) \leq 4\sqrt{\frac{2\ln(4/\delta)}{n}} +$$
$$2\sqrt{\frac{2\Phi(\gamma; d, H, L)(4H)^{L-1} \cdot \|v\|_2^2 \cdot \max_i \|x_i\|_2^2}{n}},$$

where

$$\Phi(\gamma; d, H, L) \triangleq$$
$$(He^{2\gamma} + (d-1)H\gamma^2)\left(1 + (H-1)\gamma^2 e^{2\gamma}\right)^{L-2}. \quad (4)$$

We call $\Phi(\gamma; d, H, L)$ *Basis-path measure*. Therefore, the generalization error $\epsilon_{gen}(\mathcal{G}_{\gamma,v}^{d,H,L})$ can be upper bounded by a function of Basis-path measure.

The proof depends on estimating the value of different types of paths and counting the number of different types of paths. We give the proof sketch of Theorem 2.

Proof of Theorem 2:

Step 1: If we denote $\mathcal{F}_\gamma = \{f : L \geq 2, h_0 = d, h_1 = \cdots = h_{L-1} = H, \phi(p) \leq \gamma\}$, the generalization error of a binary classification problem is

$$\epsilon_{gen}(\mathcal{G}_{\gamma,v}^{d,H,L}) \leq 2\|v\|_2^2 \mathcal{RA}(\mathcal{F}_\gamma) + 4\sqrt{\frac{2\ln(4/\delta)}{n}},$$

where $\mathcal{RA}(\cdot)$ denotes the Rademacher complexity of a hypothesis space (Wolf 2018). Following the results of Theorem 1 and Theorem 5 in (Neyshabur, Tomioka, and Srebro 2016) under $p = 2$ and $q = \infty$, we have

$$\mathcal{RA}(\mathcal{F}_\gamma) \leq \sqrt{\frac{2\Omega^2(\mathcal{F}_\gamma)(4H)^{L-1}\max_i \|x_i\|_2^2}{n}},$$

where $\Omega(\mathcal{F}_\gamma)$ is the maximal path norm of $f, f \in \mathcal{F}_\gamma$.

Step 2 (estimating path value): We give $\Omega(\mathcal{F}_\gamma)$ an upper bound using Basis-path norm. Using $\phi(p) \leq \gamma$, we have $e^{-\gamma} \leq |p_i| \leq e^\gamma$ and $|p_j| \leq \gamma$. Then using Proposition 1, we have

$$|\tilde{p}| \leq |\prod_{i=1}^{m} p_i^{\alpha_i} \prod_{j=m+1}^{r} p_j^{\alpha_j}| \quad (5)$$
$$\leq e^{-\gamma \sum_{i=1}^m \alpha_i} \cdot \gamma^{\sum_{i=m+1}^r \alpha_j}, \quad (6)$$

where $\alpha_i \leq 0, \alpha_j \geq 0$.

As shown in skeleton method in (Meng et al. 2018) (which can also be referred in appendices), basis paths are constructed according to skeleton weights. Here, we clarify the non-basis paths according to the number of non-skeleton weights it contains. We denote the non-basis path which contains b non-skeleton weights as $\tilde{p}_b$. The proofs of Proposition 1 indicates that for $\tilde{p}_b$, $\sum_{i=1}^m \alpha_i = 1 - b$, $\sum_{i=m+1}^r \alpha_j = b$. Thus we have

$$|\tilde{p}_b| \leq e^{\gamma(b-1)} \cdot \gamma^b.$$

Step 3 (counting the number of different type of paths): Based on the construction of basis paths (refer to the skeleton method in appendices), in each hidden layer, there are H skeleton weights and $H(H-1)$ non-skeleton weights. We can get that the number of $\tilde{p}_b$ in a L-layer MLP with width H is $(d-1)HC_{L-2}^{b-1}(H-1)^{b-1} + HC_{L-2}^{b}(H-1)^b$ if $1 \leq b \leq L-2$ and $(d-1)H(H-1)^{L-2}$ if $b = L-1$.

Step 4: We have:

$$\Omega^2(\mathcal{F}_\gamma) = \sum_{i=1}^{m}(p_i)^2 + \sum_{j=m+1}^{r}(p_j)^2 + \sum_{b=2}^{L-1}\sum_{k}\tilde{p}_{b,k}^2. \quad (7)$$

The number of negative basis paths is H and $e^{-\gamma} \leq |p_i| \leq e^\gamma$, so we have $\sum_{i=1}^m p_i^2 \leq He^{2\gamma}$, where $m = H$.

$$\begin{aligned}
&\Omega^2(\mathcal{F}_\gamma) \\
\leq& He^{2\gamma} + \sum_{b=1}^{L-2}\left((d-1)HC_{L-2}^{b-1}(H-1)^{b-1} + HC_{L-2}^{b}(H-1)^b\right) \\
&\cdot \left(\gamma^b e^{\gamma(b-1)}\right)^2 + (d-1)H(H-1)^{L-2} \cdot \left(\gamma^{L-1}e^{\gamma(L-2)}\right)^2 \\
\leq& (He^{2\gamma} + (d-1)H\gamma^2 \sum_{b=0}^{L-2} C_{L-2}^b (H-1)^b \left(\gamma^2 e^{2\gamma}\right)^b \\
\leq& (He^{2\gamma} + (d-1)H\gamma^2)\left(1 + (H-1)\gamma^2 e^{2\gamma}\right)^{L-2} \\
=& \Phi(\gamma; d, H, L),
\end{aligned} \quad (8)$$

where Ineq. (8) is established by the calculation of $(1+x)^a$. □

Based on the above theorem, we discuss how $\Phi(\gamma; d, H, L)$ changes as width H and depth L. (1) For fixed γ, $\Phi(\gamma; d, H, L)$ increases exponentially as L and H goes to large. (2) $\Phi(\gamma; d, H, L)$ increases as γ increases. If γ diminishes to zero, we have $\Phi(\gamma; d, H, L) \to H$. In this case, the feature directly flow into the output, which means that $f_k(x) = x_{k \mod d}$, for $k = 1, \cdots, H$. (3) If $\gamma = \mathcal{O}\left(\frac{1}{\sqrt{HL}}\right)$, we have $\Phi(\gamma; d, H, L) \leq \mathcal{O}\left(H + \frac{d}{L}\right)$. It increases linearly as d and H increase and decreases linearly as L increases.

Empirical Verification

In the previous section, we derived a BP norm induced generalization error bound for ReLU networks. In this section, we study the relationship between this bound and the empirical generalization gap - the absolute difference between test error and training error with real-data experiments, in comparison with the generalization error bounds given by other

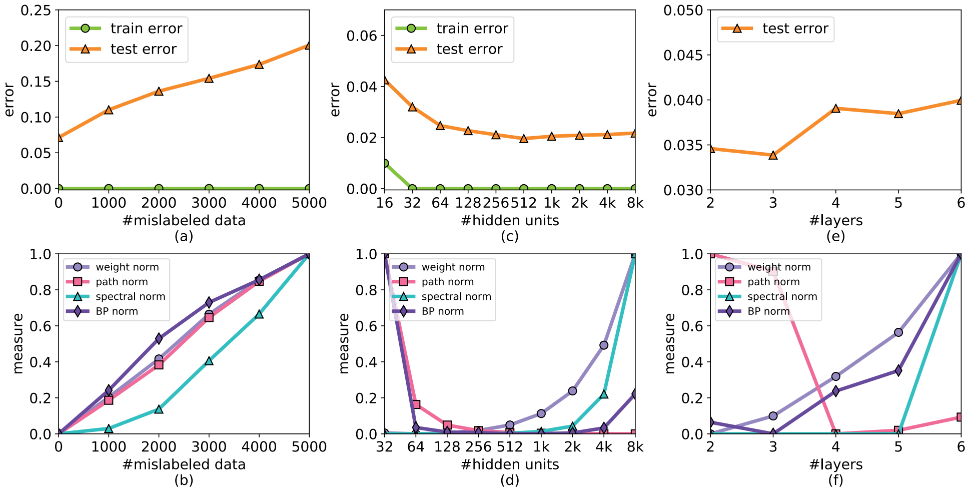

Figure 2: Left: experiments on different global minima for the objective function on the subset with true labels:(a) the training and test error, (b) different measures w.r.t. the size of random labels. Middle: experiments on different hidden units: (c) the training and test error, (d) different measures w.r.t. the size of hidden units for each layer. Right: (e) the test error, (f) different measure w.r.t. the number of layers of the network. The training errors in (e) are 0s, therefore we omit it.

capacity measures, including weight norm (Evgeniou, Pontil, and Poggio 2000), path norm (Neyshabur, Tomioka, and Srebro 2016) and spectral norm (Bartlett, Foster, and Telgarsky 2017). We follow the experiment settings in (Neyshabur et al. 2017), and extend on our BP norm bound. As shown in the previous section, the BP norm with capacity is proportional to Eqn. (4) We conduct experiments with multi-layer perceptrons (MLP) with ReLU of different depths, widths, and global minima on MNIST classification task which is optimized by stochastic gradient descent. More details of the training strategies can be found in the appendices. All experiments are averaged over 5 trials if without explicit note.

First, we train several MLP models and force them to converge to different global minima by intentionally replacing a different number of training data with random labels, and then calculate the capacity measures on these models. The training set consists of 10000 randomly selected samples with true labels and another at most 5000 intentionally mislabeled data which are gradually added into the training set. The evaluation of error rate is conducted on a fixed 10000 validation set. Figure 2 (a) shows that every network is enough to fit the entire training set regardless of the amount of mislabeled data, while the test error of the learned networks increases with increasing size of the mislabeled data. As shown in Figure 2 (b), the measure of BP norm is consistent with the behaviors of generalization on the data and indeed is a good predictor of the generalization error, as well as weight norms, path norm, and spectral norm.

We further investigate the relationship between generalization error and the network size with different widths. We train a bunch of MLPs with 2 hidden layers and varying number of hidden units from 16 to 8192 for each layer. The experiment is conducted on the whole training set with 60000 images. As shown in Figure 2(c), the networks can fit the whole training set when the number of hidden units is greater than or equal to 32, while the minimal test error is achieved with 512 hidden units, then shows a slightly over fitting on training set beyond 1024 hidden units. Figure 2(d) shows that the measure of BP norm behaves similarly to the trend of generalization errors which decreases at the beginning and then slightly increases, and also achieves minimal value at 512 hidden units. Weight norm and spectral norm keep increasing along with the network size growing while the trend of generalization error behaves differently. Path norm shows the good explanation of the generalization when the number of hidden units is small, but keeps decreasing along with increasing the network size in this experiment. One possible reason is that the proportion of basis paths in all paths is decreasing, and the vast majority improperly affects the capacity measure when the dependency in the network becomes large. In contrast, BP norm better explains the generalization behaviors regardless of the network size.

Similar empirical observation is shown when we train the network with a different number of hidden layers. Each network has 32 hidden units in each layer and can fit the whole training set in this experiment. As shown in Figure2(e,f), the minimal test error is achieved with 3 hidden layers, and then shows an over fitting along with the increasing of the layers. The weight norm keeps increasing with the growing of network size as discussed above, and the $\Pi_i h_i$ in spectral norm will be quite large when layers L is increasing. Path norm can partially explain the decreasing generalization error before 4 hidden layers and it indicates that the networks with 4, 5 and 6 hidden layers have small generalization error, which doesn't match our observations. The amount of non-basis paths is exponentially growing when layers L is increasing, therefore the path norm couldn't measure the capacity accurately by counting all paths' values. In contrast, the BP norm

Algorithm 1 Optimize ReLU Network with SGD and Basis-path Regularization

```
Require: learning rate $\eta_t$, training set $S$, initial $w_0$.
  for $t = 0, \cdots, T$ do
      1. Draw mini-batch data $x^t$ from $S$.
      2. Compute gradient of the loss function $g(w^t) = \nabla f(w^t, x^t)$.
      3. Compute gradient of the basis-path regularization $h(w^t) = \nabla R(w)$ by Eqn. (10) and (11).
      4. Update $w^{t+1} = w^t - \eta_t(g(w^t) + h(w^t))$.
  end for
```

can nearly match the generalization error, these observations verify that BP norm bound is more tight to generalization error and can be a better predictor of generalization.

Basis-path Regularization for ReLU Networks

In this section, we propose Basis-path regularization, in which we penalize the loss function by the BP norm. According to the definition of BP norm in Eqn.(2), to make it small, we need to restrict the values of negative basis paths to be moderate (neither too large nor too small) and minimize the value of positive basis paths. To this end, in our proposed method, we penalize the empirical loss by the l_2 distance between the values of negative basis paths and 1, as well as the sum of the values of all positive basis paths.

The constraint $\phi(p) \leq \gamma$ equals to $\|p^+\|^2 \leq \gamma^2$ and $\|\log(p^-)^2\|^2 \leq (2\gamma)^2$, which means that the largest element in a vector is smaller than γ iff all of the element is smaller than γ. We choose to optimize their square because of the smoothness. So using the Lagrangian dual methods, we add the constraint $\frac{\lambda_1}{2}\|p^+\|^2$ and $\frac{\lambda_2}{4}\|\log(p^-)^2\|^2$ in the loss function and then optimize the regularized empirical risk function:

$$\begin{aligned} L(w,x) &= f(w,x) + R(p) \\ &= f(w,x) + \frac{\lambda_1}{2}\|p^+\|^2 + \frac{\lambda_2}{4}\|\log(p^-)^2\|^2. \end{aligned} \tag{9}$$

We use $g(w)$ to denote the gradient of loss with respect to w, i.e., $g(w) = \frac{\partial f(w,x)}{\partial w}$. For the non-skeleton weight w_j, since it is contained in only one positive basis path p_j, we can calculate the gradient of the regularization term with respect to w_j as

$$h(w_j) = \frac{\lambda_1}{2}\frac{\partial R(p)}{\partial p_j}\frac{\partial p_j(w)}{\partial w_j} = \lambda_1 \cdot \frac{p_j^2}{w_j}. \tag{10}$$

For the skeleton weight w_i, it is contained in only one negative basis path p_i (if the neural network has equal number of hidden nodes) and some of the positive basis paths p_j. Thus its gradient can be calculated as follows

$$\begin{aligned} h(w_i) &= \frac{\lambda_2}{4}\frac{\partial R(p)}{\partial p_i}\frac{\partial p_i(w)}{\partial w_i} + \frac{\lambda_1}{2}\sum_{p_j:w_i}\frac{\partial R(p)}{\partial p_j}\frac{\partial p_j(w)}{\partial w_i} \\ &= \frac{\lambda_2 \log p_i}{w_i} + \lambda_1 \sum_{p_j:w_i}\frac{p_j^2}{w_i}, \end{aligned} \tag{11}$$

where $p_j : w_i$ denotes all positive basis paths containing w_i.

Combining them together, we get the gradient of the regularized loss function with respected to the weights. For example, if we use stochastic gradient descent to be the optimizer, the update rule is as follows:

$$w^{t+1} = w^t - \eta_t(g(w^t) + h(w^t)). \tag{12}$$

Please note that the computation overhead of $h(w_i)$ is high, moreover, we observed that the values of the negative basis paths are relatively stable in the optimization, thus we set $h(w_i)$ to be zero for ease of the computation. Specifically, basis-path regularization can be easily combined with the optimization algorithm which is in quotient space.

The flow of SGD with basis-path regularization is shown in Algorithm 1, it's trivial to extend basis-path regularization to other stochastic optimization algorithms. Comparing to weight decay, basis-path regularization has little additional computation overhead. All the additional computations regarding Eqn.(10) only introduce very lightweight element-wise matrix operations, which is small compared with the forward and backward process.

Experimental Results

In this section, we evaluate Basis-path Regularization on deep ReLU neural networks with the aim of verifying that does our proposed BP regularization outperforms other baseline regularization methods and whether it can improve the generalization on the benchmark datasets. For sake of fairness, we reported the *mean* of 5 independent runs with random initialization.

Recommendation System

We first apply our basis-path regularization method to recommendation task with MLP networks and conduct experimental studies based on a public dataset, MovieLens[3]. The characteristics of the MovieLens dataset are summarized in Table 1. We use the version containing one million ratings, where each user has at least 20 ratings. We train an NCF framework with similar MLP network proposed in (He et al. 2017) and followed their training strategies with Adam optimizer but without any pre-training. We test the *predictive factors* of [8,16,32,64], and set the number of hidden units to the embedding size ×4 in each hidden layer. We calculate both metrics for each test user and report the average score. For each method, we perform a wide range grid search of hyper-parameter λ from $10^{-\alpha}$ where $\alpha \in 5, 6, 7, 8, 9$ and report the experimental results based on the best performance on the validation set. The performance of a ranked list is judged by *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain* (NDCG) (He et al. 2015).

Figure 3 (a) and (b) show the performance of HR@10 and NDCG@10 w.r.t. the number of predictive factors. From this figure, it's clear to see that basis-path regularization achieve better generalization performance than all baseline methods. Figure 3 (c) and (d) show the performance of Top-K recommended lists where the ranking position K ranges from 1

[3] https://grouplens.org/datasets/movielens/1m/

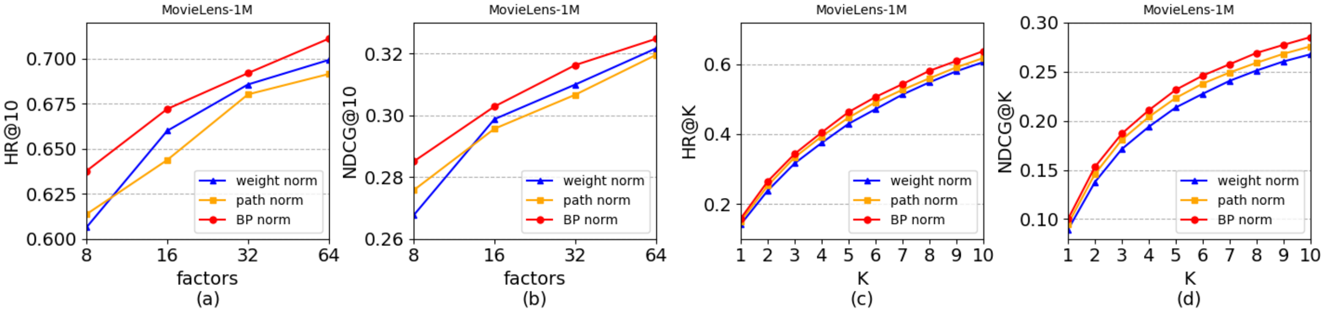

Figure 3: Performance of HR and NDCG w.r.t. the number of predictive factors and Top-K items recommendation.

Table 1: Statistics of the MovieLens datasets.

Dataset	Interaction#	Item#	User#	Sparsity
MovieLens	1,000,209	3,706	6,040	95.53%

to 10. As can be seen, the basis-path regularization demonstrates consistent improvement over other methods across positions, which is consistent with our analysis of generalization error bound in the previous section.

Image Classification

In this section, we apply our basis-path regularization to this task and conduct experimental studies based on CIFAR-10 (Krizhevsky and Hinton 2009), with 10 classes of images. We employ a popular deep convolutional ReLU model, ResNet (He et al. 2016) for image classification since it achieves huge successes in many image related tasks. In addition, we conduct our studies on a stacked deep CNN described in (He et al. 2016) (refer to PlainNet), which suffers serious dependency among the paths. We train 34 layers ResNet and PlainNet networks on this dataset, and use SGD with widely used l_2 weight decay regularization (WD) as our baseline. In addition, we implement $\mathcal{G}$-SGD, which is proposed in (Meng et al. 2018) and optimize the networks on basis paths. We investigate the combination of SGD/$\mathcal{G}$-SGD and basis-path regularization (BPR). Similar with the previous task, we perform a wide range grid search of λ from $\{0.1, 0.2, 0.5\} \times 10^{-\alpha}$, where $\alpha \in \{3, 4, 5, 6\}$. More training details can be found in supplementary materials.

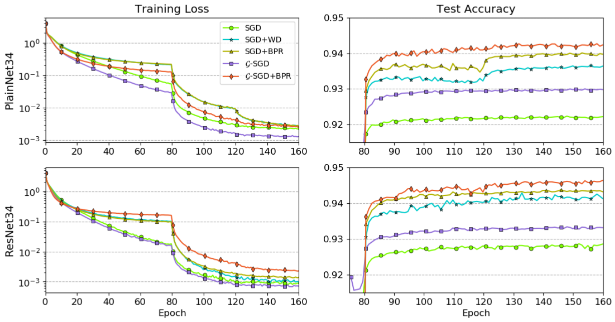

Figure 4: Training loss and test accuracy of the PlainNet34 and ResNet34 models w.r.t. number of effective passes on data.

Table 2: Classification error rate (%) on image classification task. Baseline is from (He et al. 2016), and the number of † is 7.51 reported in the original paper. Fig. 4 shows the training procedures.

Algorithm	PlainNet34			ResNet34		
	Train	Test	Δ	Train	Test	Δ
SGD	0.06	7.76	7.70	0.01	7.13	7.12
SGD + WD	0.06	6.34	6.27	0.01	5.71†	5.70
SGD + BPR	0.06	**5.99**	**5.92**	0.01	**5.62**	**5.61**
$\mathcal{G}$-SGD	0.03	7.00	6.97	0.01	6.66	6.65
$\mathcal{G}$-SGD + BPR	0.05	**5.73**	**5.68**	0.03	**5.36**	**5.33**

Figure 4 and Table 2 shows the training and test results of each algorithms. From the figure and table, we can see that our basis-path regularization indeed improves test accuracy of PlainNet34 and Resnet34 by nearly 1.8% and 1.5% respectively. Moreover, the training behaviors of SGD with weight decay and basis-path regularization are quite similar, but the basis-path regularization can always find better generalization points during optimization, which is consistent with our theoretical analysis in the previous section. We further investigate the combination of $\mathcal{G}$-SGD and basis-path regularization. $\mathcal{G}$-SGD with basis-path regularization achieves the best test accuracy on both PlainNet and ResNet model, which indicates that taking BP norm as the regularization term to the loss function is helpful for optimization algorithms.

Conclusion

In this paper, we define Basis-path norm on the group of basis paths, and prove that the generalization error of ReLU neural networks can be upper bounded by a function of BP norm. We then design Basis-path regularization method, which shows clearly performance gain on generalization ability. For future work, we plan to test basis-path regularization on larger networks and datasets. Furthermore, we are also interested in applying basis-path regularization on networks with different architecture.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant U1636201.

We would like to show our gratitude to Prof. Zhi-Ming Ma, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, for sharing his pearls of wisdom and providing his patient guidance with us during the course of this research.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

Bartlett, P. L.; Foster, D. J.; and Telgarsky, M. J. 2017. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, 6241–6250.

Chan, W.; Jaitly, N.; Le, Q.; and Vinyals, O. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 4960–4964. IEEE.

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 7–10. ACM.

Dziugaite, G. K., and Roy, D. M. 2017. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*.

Evgeniou, T.; Pontil, M.; and Poggio, T. 2000. Regularization networks and support vector machines. *Advances in computational mathematics* 13(1):1.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323.

He, X.; Chen, T.; Kan, M.-Y.; and Chen, X. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1661–1670. ACM.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, 173–182. International World Wide Web Conferences Steering Committee.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference of Machine Learning*.

Kawaguchi, K.; Kaelbling, L. P.; and Bengio, Y. 2017. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*.

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

Meng, Q.; Zheng, S.; Zhang, H.; Chen, W.; Ma, Z.-M.; and Liu, T.-Y. 2018. G-sgd: Optimizing relu neural networks in its positively scale-invariant space. *arXiv preprint arXiv:1802.03713*.

Neyshabur, B.; Wu, Y.; Salakhutdinov, R. R.; and Srebro, N. 2016. Path-normalized optimization of recurrent neural networks with relu activations. In *Advances in Neural Information Processing Systems*, 3477–3485.

Neyshabur, B.; Bhojanapalli, S.; McAllester, D.; and Srebro, N. 2017. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, 5949–5958.

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

Neyshabur, B.; Tomioka, R.; and Srebro, N. 2015. In search of the real inductive bias: On the role of implicit regularization in deep learning. *International Conference on Learning Representations*.

Neyshabur, B.; Tomioka, R.; and Srebro, N. 2016. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, 1376–1401.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 6000–6010.

Wolf, M. M. 2018. Mathematical foundations of supervised learning. *[J]*.

Xiong, W.; Droppo, J.; Huang, X.; Seide, F.; Seltzer, M.; Stolcke, A.; Yu, D.; and Zweig, G. 2016. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. *ICLR*.