WASSERSTEIN GENERALIZATION BOUND FOR FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In the absence of large quantities of annotated data, few shot learning is used to train neural networks that make predictions based on similarities between datapoints. To better understand how models would behave when presented with unfamiliar data, research on generalization bounds have revealed some important properties about deep neural networks. However, when extended to the domain of few shot learning it often yields loose bounds since it does not take into the account the nature and methodology of few shot learning. We propose a novel stochastic generalization bound for prototypical neural networks by constructing a Wasserstein sphere centered around the distribution of weight matrices. We show that by applying concentration inequalities on the distribution of weight matrices in the Wasserstein sphere stricter generalization bounds can be obtained. Comparison with previous generalization bounds shows the efficacy of our approach and to our knowledge this is the first bound that makes use of Wasserstein distance to give a measure of generalizability of deep neural networks.

1 INTRODUCTION

The problem of finding sharp generalization bounds for deep neural networks is of prominent importance as it allows us to bound the overall uncertainty involved in their application. In recent times the theoretical properties of these bounds have received increased attention and has been an active subject of investigation. Various classical results exploring the r expressivity of neural networks have acknowledged their universality Leshno et al. (1993) and their unexpected advantage over-hand crafted features Barron (1993) even though training of neural networks itself is a hard problem Blum & Rivest (1992). Other studies have also revealed that deep neural networks may have structural properties that enable them to perform non-convex optimization Choromanska et al. (2015); Kawaguchi (2016) further alluding to the fact that given enough data these models can learn any function Cybenko (1989). However, simply possessing such desirable properties does not guarantee that the models will perform accurately on future unknown inputs, this is because without proper restrictions on the optimization the models become prone to over-fitting and to effectively address this challenge leads us to study the generalization of these models. However, though there exits a great body of research pertaining the generalization of classification models relatively little is studied about generalization properties of meta learning models Vanschoren (2019), specifically Few-shot learning (FSL) Wang et al. (2020).

In this paper we study the generalization of FSL specifically that of Prototypical Networks Snell et al. (2017). By leveraging stochastic bounds from classic PAC learning theory Vapnik et al. (1994) we derive a Wasserstein bound on the probability of the absolute difference between the true and the empirical error deviating from a established threshold. Some of the most sharp generalization bounds are obtained using the PAC-Bayesian Framework McAllester (1998; 1999) and in this work we make use of it to derive a stochastic bound for FSL involving Prototypical networks. However, the standard PAC-Bayesian framework relies

on the KL divergence between some prior distribution of set of classifiers and data distribution, our work leverages the Wasserstein metric Vallender (1974) to obtain a better bound. The unique nature of FSL is in stark contrast to traditional task of classification and when combining them with the methodology used in classification of prototypical networks we are able to obtain a tight stochastic bound.

Also prior works assume homogeneous nature of data samples while obtaining the bound , we however do not impose any such restriction while studying it and also our final bound involves the deviation of final distribution from the initial distribution in wasserstein metric.

2 Related Work

Various classical theory work attributes generalization ability to understanding the class-capacity Vapnik (1999); Mohri et al. (2018). Recent work in deep hypothesis spaces Pascanu et al. (2013); Montufar et al. (2014); Livni et al. (2014); Telgarsky (2016) also revealed deep neural networks can perform convex optimization thereby being to generalize over a vast set of datapoints. Harvey et al. (2017) generalization error bound showed that the VC dimension of neural network depends on the product of it depth and parameters considerably improving the previous bounds given by Bartlett et al. (1998). Feed forward neural networks were revealed to have unit-wise ℓ_1 norm generalization bound with exponential dependence on depth. A sharpness based measure was suggested by Keskar et al. (2016) to predict the difference in generalization behaviours of networks trained with different batch size SGD. More recent PAC-Bayesian approaches Neyshabur et al. (2017); Nagarajan & Kolter (2019) also gave very sharp bounds utilizing spectral and Frobenius norm of weight matrices. In the domain of few shot learning Cao et al. (2019) provided a framework to obtain the optimal *k* shot for prototypical networks.

3 Background

3.1 Problem Setup

Consider N distinct classes being sampled i.i.d. from the set of all possible classes $\mathcal{C}$ for an N-way classification problem. For each class $c_i \in \{c_1, c_2, \ldots, c_N\}$ k datapoints are sampled i.i.d. from the class conditional distribution $p(x|Y(x) = c_i)$, where $x \in \mathbb{R}^D$, $Y(x)$ is the class assignment of x and D is the dimension of the data.

The k datapoints constitute the support set of the class $c_i : S_i = \{x_1, \ldots x_k\}$ where $Y(x_j) = c_i$ and $x_j \in S_i$ for all $c_i \in \mathcal{C}$. Given a datapoint (x_j, y_j), where $y_j \in \{c_1, c_2, \ldots, c_N\}$ and $x_j \notin \{S_1, \ldots, S_k\}$, the few shot classification task is to predict the correct assignment label y_j using $\mathcal{S} = \bigcup_{i=1}^{N} S_i$.

3.2 Prototypical Networks

Prototypical Networks Snell et al. (2017) are trained to learn the low dimensional representation of data i.e., they learn a function $\phi : \mathbb{R}^D \to \mathbb{R}^M$, where M is the dimension of the representation space. The prototype representation of each class $\phi(S_i)$ is generating by taking the average of the representations of its support set:

$$\phi(S_i) = \frac{1}{k} \sum_{x \in S_i} \phi(x) \tag{1}$$

Classification of input x is obtained by taking the softmax of the distance between the input embedding and the prototype representation of each class:

$$p_\phi(y = j|x, \mathcal{S}) = \frac{\exp\left(-d\left(\phi\left(x\right), \phi\left(S_j\right)\right)\right)}{\sum_{i=1}^{N} \exp\left(-d\left(\phi\left(x\right), \phi\left(S_i\right)\right)\right)} \tag{2}$$

where d is a distance function : $\mathbb{R}^M \times \mathbb{R}^M \to [0, +\infty)$.

Most applications including our approach use the euclidean distance as the distance function. Learning involves minimizing the negative log probability $J(\phi) = -\log p_\phi(\hat{y} = j|x)$ using SGD and the function ϕ is generally a deep neural network.

3.3 Wasserstein Ball and Total Variation

3.4 Wasserstein Ball

Given the space of all probability distributions $\mathcal{P}$ with compact support set, the p^{th} Wasserstein metric on the space $\mathcal{P}$ is defined as:

$$W_p(\nu, \mu) = (\inf \mathbb{E}[d(X, Y)^p])^{1/p} \tag{3}$$

where X and Y are random variables with marginals μ and v and infimum is taken over all possible joint distributions of X and Y. For our analysis we focus on the first order Wasserstein distance by taking the distance measure d as the Manhattan distance:

$$W(\nu, \mu) = \Big(\inf \ \mathbb{E}\big[\ \|(X, Y)\|_1 \ \big] \Big) \tag{4}$$

The rationale for using Wasserstein distance is that it gives a metric to measure the minimum difference between two distributions which we use to obtain a sharper generalization bound. Consequently, a wasserstein ball of radius R centered around μ is defined as:

$$\mathcal{W}_\mu(R) = \big\{ v \in \mathcal{P} \ \big| \ W(\nu, \mu) \leq R \big\} \tag{5}$$

3.5 Total Variation

The total variation distance between two distributions ν and μ is given by

$$\delta(\nu, \mu) = \sup_{A \in \mathcal{P}} |\nu(A) - \mu(A)| \tag{6}$$

Intuitively, this is the largest possible difference between the probabilities that the two probability distributions can assign to the same event. In some cases we can also have the below relation

$$\delta(\nu, \mu) = \frac{1}{2} ||\nu - \mu||_1 \tag{7}$$

This is similar to Wasserstein distance in many aspects , The total variation distance (or half the norm) arises as the optimal transportation cost, when the cost function is $c(x, y) = \mathbf{1}_{x \neq y} c(x, y) = \mathbf{1}_{x \neq y}$, that is,

$$\frac{1}{2} ||\nu - \mu||_1 = \delta(\nu, \mu) = \inf \mathcal{P}(\nu \neq \mu) = \inf_{\pi} \mathbb{E}_\pi[\mathbf{1}_{\nu \neq \mu}] \tag{8}$$

It however differs in taking distributions directly rather than their supports , which makes it less desirable than wasserstein distance for our case. We use it mainly to compare the wasseserstein diatnce with the $K - L$ divergence , which otherwise cannot be compared mathematically with wasserstein metric.

4 WASSERSTEIN BOUND

Prototypical networks make predictions are based on the nearest neighbour from the support set $\mathcal{S}$ and the embedding function ϕ. Thus, to formulate the relationship between the complexity of the classifier and the support set we make us of the classical PAC learning theory Vapnik et al. (1994). Consider the simple binary classification problem where the probability of the difference between true and empirical error is bounded by:

$$P\bigg(||err_{true}(h) - err_{train}(h)||_1 \leq \xi\bigg) \geq 1 - \delta \tag{9}$$

where h is the classifier, err_{true} is true error, err_{train} is the empirical training error obtained on the support set $\mathcal{S}$, $0 \leq \delta \leq 1$ and ξ is defined as:

$$\xi \triangleq \sqrt{\frac{D\left(\ln \frac{4k}{D} + 1\right) + \ln \frac{4}{\delta}}{2k}} \tag{10}$$

where D is the VC Dimension. l_1 metric is conventionally used to measure the deviation but metrics which do not over estimate are crucial for the accurate prediction of the error difference. A sharper bound which is symmetric about the distributions is extremely necessary for studying generalization in Few shot learning.

Li showed that effective prediction by neural networks is generally the result of the final layers of the network where the embeddings are split apart to facilitate effective linear classification. However, the embeddings learnt by the networks before the final layer can be very compact in some high dimensional vector space. Consider two distributions ν and μ from this compact embeddings, the KL divergence of these two distributions is given by:

$$KL(\nu||\mu) = \int_x \mu(x) \ln\left(\frac{\nu(x)}{\mu(x)}\right) \tag{11}$$

If the context of few shot learning the embeddings $\phi(S_i)$ may be close enough enough such that their class conditional distributions are very similar, i.e.

$$\begin{aligned} KL(\nu||\mu) &= \lim_{\nu \longrightarrow \mu} KL(\nu||\mu) \\ &= \lim_{\nu \longrightarrow \mu} \int_x \mu(x) \ln\left(\frac{\nu(x)}{\mu(x)}\right) \end{aligned} \tag{12}$$

By monotone convergence theorem, for finite measures equation (12) can be written as:

$$KL(\nu||\mu) = \int_x \lim_{\nu \longrightarrow \mu} \mu(x) \ln\left(\frac{\nu(x)}{\mu(x)}\right) = 0 \tag{13}$$

As we could see from Equation (13) the KL divergence could be pretty inaccurate in capturing the distance between the class conditional distributions tends to 0 which would be further exacerbated by the log factor present in its formulation. The Wasserstein metric is preferable in this regard to Kullback–Leibler (KL) divergence as it over comes this problem of magnitude reduction by projecting it into higher dimensional product measure space and effectively capturing it Otto & Villani (2000):

$$W(\nu, \mu) = \sqrt{\inf_{\pi \in \prod(\nu, \mu)} \int_{M \times M} d(x, y)^2 d\pi(x, y)} \tag{14}$$

where $\prod(\nu, \mu)$ denotes the set of probability measures on $M \times M$ where M is some finite dimensional vector space. More specifically, for any two distributions ν and μ by Equation (11) and (14) we have the following

inequalities demonstrating the sharpness of the Wasserstein metric in comparison to KL-divergence in the limiting cases:

$$\frac{1}{2}d_{TV}(\nu,\mu) < \sqrt{KL(\nu,\mu)} \tag{15}$$

$$W(\nu,\mu) \leq \mathcal{O}(d_{TV}(\nu,\mu)) \tag{16}$$

From Equations (15) and (16) we can conclude that $W(\nu,\mu) \leq \sqrt{KL(\nu,\mu)}$. Therefore, the usage of a was a Wasserstein distance is more appropriate in the present context of few shot generalization.

Lemma 1. *Given a prototypical network ϕ_θ with a N-way k-shot classification task, a query sample $x_q \in \mathbb{R}^D$ with support set $\mathcal{S}$,R is the radius of this support set and if R_i is the radius of the Wasserstein ball for class c_i centered around $\phi(S_i)$, for $Z_i = \phi(x_q) - \overline{\phi(S_i)}$, we define $v(Z_i) = ||E(Z_i.Z_i^*||$, then $v(Z_i)$ can be simplified as*

$$v(Z) = 2(1+\frac{1}{k})(R+R_i) \tag{17}$$

Proof. First, from the definition of $v(Z)$, by taking conditional probabilities into account we get $v(Z) = \mathbb{E}_{\mathbf{x},S_i}$, since $x_q \in \mathbb{R}^D$ hence $\overline{\phi(x)} = \phi(x)$ into two parts and examine them separately:

$$v(Z_i) = \mathbb{E}_{\mathbf{x},S_i} = \mathbb{E}[(\phi(x) - \overline{\phi(S_i)}).(\phi(x) - \phi(S_i))] \tag{18}$$

In general, from probability theory we have for random vector X, the expectation of the quadratic is $\mathbb{E}[||X||^2] = \text{Tr}(\text{Var}(X)) + \mathbb{E}[X]^T\mathbb{E}[X]$. Hence,

$$\begin{aligned} v(Z_i) &= \mathbb{E}[||\phi(x) - \overline{\phi(S_i)}||^2] \\ &= \text{Tr}(\Sigma_{\phi(x)-\overline{\phi(S_i)}}) + \mathbb{E}[\phi(x) - \overline{\phi(S_i)}]^*\mathbb{E}[\phi(x) - \overline{\phi(S_i)}], \end{aligned} \tag{19}$$

where the first term inside the trace can be expanded as:

$$\begin{aligned} \Sigma_{\phi(x)-\overline{\phi(S_i)}} &= \text{Var}[\phi(x) - \overline{\phi(S_i)}] \\ &= \mathbb{E}[(\phi(x) - \overline{\phi(S_i)})(\phi(x) - \overline{\phi(S_i)})^T] - (\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^T \\ &= \Sigma_c + \boldsymbol{\mu}_a\boldsymbol{\mu}_a^T + \frac{1}{k}\Sigma_c + \boldsymbol{\mu}_b\boldsymbol{\mu}_b^T - \boldsymbol{\mu}_a\boldsymbol{\mu}_b^T - \boldsymbol{\mu}_b\boldsymbol{\mu}_a^T - (\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^T \\ &= (1+\frac{1}{k})\Sigma_c \quad \text{(Last terms cancel out).} \end{aligned} \tag{20}$$

by linearity of trace we can obtain the following equation from equation(20)

$$\text{Trace}(\Sigma_{\phi(x)-\overline{\phi(S_i)}}) = (1+\frac{1}{k})\text{Trace}(\Sigma_c) \tag{21}$$

We note that $\text{Var}(X) = \mathbb{E}[XX^T] - \mathbb{E}[X]\mathbb{E}[X]^T$ and $\Sigma_c \triangleq \text{Var}(\phi(x) + \phi(S_i))$. Hence, equation (20) is obtained by expanding out the first term and taking the expectation of each resulting item. The second term of Equation (19) is rewritten for notational convenience as :

$$\mathbb{E}_{\mathbf{x},S}[||\phi(x) - \overline{\phi(S_i)}||^2] = \boldsymbol{\mu}_a - \boldsymbol{\mu}_b. \tag{22}$$

Putting them together:

$$i = (1+\frac{1}{k})\text{Tr}(\Sigma_c) + (\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^T(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b) \tag{23}$$

□

Similarly for $\mathbb{E}[\phi(x) - \overline{\phi(S_i)}]^* \mathbb{E}[\phi(x) - \overline{\phi(S_i)}]$ we have :

$$\begin{aligned}\mathbb{E}[\phi(x) - \overline{\phi(S_i)}]^* \mathbb{E}[\phi(x) - \overline{\phi(S_i)}] &= \mathbb{E}_{\mathbf{x},S}[||\phi(x) - \overline{\phi(S_i)}||^2] \\ &= \text{Tr}(\Sigma_{\phi(x)-\overline{\phi(S_i)}}) + \mathbb{E}[\phi(x) - \overline{\phi(S_i)}]^* \mathbb{E}[\phi(x) - \overline{\phi(S_i)}] \\ &= (1 + \frac{1}{k})\text{Tr}(\Sigma_c).\end{aligned} \tag{24}$$

Putting together equation21 and equation24:

$$\begin{aligned}\mathbb{E}_{\mathbf{x},S} &= (1 + \frac{1}{k})\text{Tr}(\Sigma_c) + (\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^T + (1 + \frac{1}{k})\text{Tr}(\Sigma_c) \\ &= (\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^T(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b) + 2.(1 + \frac{1}{k})\text{Tr}(\Sigma_c).\end{aligned} \tag{25}$$

we note that $\boldsymbol{\mu}_a^T \boldsymbol{\mu}_a$ and $\boldsymbol{\mu}_b^T \boldsymbol{\mu}_b$ are quadratic forms while $\boldsymbol{\mu}_a^T \boldsymbol{\mu}_b$ describe the dot product between two independent randomly drawn samples which has expectation 0 , as we assume all the random variables involved are centered around 0. By the iid assumption on the random variables , off-diagonal terms of the Σ are zero , hence trace is just the variance of the random vector .

Variance is however the largest possible deviation in the distributed space hence by the assumptions made in the lemma we can write the final expression as $2(1 + \frac{1}{k})(R + R_i)$

For proof of Lemma 1, we first re-state the result on quadratic forms of normally distributed random vectors by Rencher & Schaalje (2008).

Theorem 2. *Given a prototypical network ϕ_θ with a N-way k-shot classification task, a query sample $x_q \in \mathbb{R}^D$ with support set $\mathcal{S}$, comes from a sphere of radius R, then the probability the model correctly predicts the class assignment bounded by:*

$$p_\phi(y = j|x_q, \mathcal{S}) \leq \prod_{i=1}^{N} 1-(1+D)\left[\exp\left(-\frac{3}{2}\left(\frac{(||\phi_\theta(x_q)||_2 - R_i)^2}{3.2.(1 + \frac{1}{k})(R + R_i) - L\,(||\phi_\theta(x_q)||_2 - R_i)^2}\right)\right)\right] \tag{26}$$

where R_i is the radius of the Wasserstein ball centered around $\phi(S_i)$ measured in wasserstein metric and includes class c_i ,i.e $c_i \in \mathcal{W}_{\phi(S_i)}(R_i)$and $L = \max(R_1, \ldots, R_N)$, $Y(x_q) = j$

Proof. The Wasserstein ball for each class c_i is given by equation (5):

$$\mathcal{W}_{\phi(S_i)}(R_i) = \left\{v \in \mathcal{P}_{c_i} \;\middle|\; W(\mu, v) \leq R_i\right\} \tag{27}$$

where $\mathcal{P}_{c_i}$ is the class conditional distribution. For the prototypical network ϕ_θ to correctly predict $Y(x_q)$ the representational embedding of x_q must be closer to $\phi(S_j)$, i.e. $\phi(x_q)$ should be closer to the center of the Wasserstein ball $\mathcal{W}_{\phi(S_j)}$ than any other $\mathcal{W}_{\phi(S_i)}$ for all $m \in \{1, \ldots, N\}$ and $j \neq m$:

$$p_\phi(y = j|x_q, \mathcal{S}) \geq p_\phi(y = m|x_q, \mathcal{S}) \tag{28}$$

For the network to generalize to previously unseen query samples equation (28) should hold true. Therefore:

$$\exp\left(-d\left(\phi\left(x\right), \phi\left(S_j\right)\right)\right) \geq \exp\left(-d\left(\phi\left(x\right), \phi\left(S_m\right)\right)\right) \tag{29}$$

Since the classification depends only on the distance between the representational embeddings, Equation (29) can we rewritten as:

$$P\Big(||\phi(x_q) - \phi(S_j)|| \geq ||\phi(x) - \phi(S_p)||\Big) \tag{30}$$

for $p = 1, 2, ...N$

As $\phi(x), S_1, \ldots, S_N$ are random we will consider expected value rather than the exact random value , now we get

$$P(||\phi(x) - E\phi(S_j)||) \geq P(||\phi(x) - E\phi(S_p)||) \tag{31}$$

Now, the probability that x_q is correctly classified by the model is given by:

$$\begin{aligned} P(y = j|x_q, \mathcal{S}) = P(&||\phi(x) - \phi(S_j)|| > ||\phi(x) - \phi(S_1)||, \\ &||\phi(x) - \phi(S_j)|| > ||\phi(x) - \phi(S_2)||, \\ &\vdots \\ &||\phi(x) - \phi(S_j)|| > ||\phi(x) - \phi(S_N)||) \end{aligned} \tag{32}$$

Next we note that that the sampling was i.i.d so we can split the RHS of Equation (32) into product of several probabilities:

$$\begin{aligned} P(y = j|x_q, \mathcal{S}) = &P(||\phi(x) - \phi(S_j)|| \geq ||\phi(x) - \phi(S_1)||) \\ &P(||\phi(x) - \phi(S_j)|| \geq ||\phi(x) - \phi(S_2)||) \\ &\vdots \\ &P(||\phi(x) - \phi(S_j)|| \geq ||\phi(x) - \phi(S_N)||) \end{aligned} \tag{33}$$

Applying Bernstein's inequality to the i^{th} probability in the product of probabilities in Equation 33 i.e., on $P(||\phi(x) - \phi(S_j)|| \geq ||\phi(x) - \phi(S_i)||)$ we get the following bound:

$$\begin{aligned} &P(||\phi(x) - \phi(S_j)|| \geq ||\phi(x) - \phi(S_i)||) \leq \\ &1 - (1 + D)\left[\exp\left(-\frac{3}{2}\left(\frac{(||\phi_\theta(x_q)||_2 - R_i)^2}{3v(Z) - L\,(||\phi_\theta(x_q)||_2 - R_i)^2}\right)\right)\right] \end{aligned} \tag{34}$$

where $v(Z) = E[\phi(x) - \phi(S_j)(\phi(x) - \phi(S_j)^*]$ and $L$is a quantity which bounds all the random embeddings of the classes in embedding space , (i.e) $L \geq ||\phi(S_i)||$, we hence choose $L = \max(R_1, \ldots, R_N)$, as all the embeddings lie in the sphere of radius R_i this quantity bounds all of them. Also , by triangle inequality we have

$$L \geq ||\phi(S_i) - \phi(S_i)|| \geq ||\phi(S_i)|| - ||\phi(S_i)|| \tag{35}$$

After applying the basic assumptions that query sample is uncorrelated with the support sets S_i , we can now use lemma(1) to further simplify this to

$$v(Z) = 3.2.(1 + \frac{1}{k})(R + R_j) \tag{36}$$

Now by using equation(35) and equation(36)we can rewrite Equation (34) as:

$$\begin{aligned} &P(||\phi(x) - \phi(S_j|| \geq ||\phi(x) - \phi(S_1||)) \leq \\ &1 - (1 + D)\left[\exp\left(-\frac{3}{2}\left(\frac{(||\phi_\theta(x_q)||_2 - R_i)^2}{3.2.(1 + \frac{1}{k})(R + R_i) - L\,(||\phi_\theta(x_q)||_2 - R_i)^2}\right)\right)\right] \end{aligned} \tag{37}$$

now in similar way applying this for all probabilities in Equation 32 factors we get the final expression of Equation (38).

$$p_\phi(y=j|x_q,\mathcal{S}) \leq \prod_{i=1}^{N} 1-(1+D)\left[\exp\left(-\frac{3}{2}\left(\frac{(||\phi_\theta(x_q)||_2-R_i)^2}{3.2.(1+\frac{1}{k})(R+R_i)-L\,(||\phi_\theta(x_q)||_2-R_i)^2}\right)\right)\right] \quad (38)$$

□

5 EXPERIMENTS

In this section, we present our result illustrating the advantage of our bound on following datasets: Omniglot Lake et al. (2015), *mini*ImageNet Vinyals et al. (2016) and *tiered*ImageNet Ren et al. (2018). In table (1) all experiments are performed on a 4 layer neural network, similar to that used by Snell et al. and 7 layer residual neural network He et al. (2016). For the purpose of clarity the specific architecture of the neural networks and the preprocessing of the data is the same as that used by Cao et al. (2019). Relatively simple models are used to highlight the behaviour of our stochastic bound given different network architecture and difference in testing shots $k \in \{1, \cdots, 5\}$. PCA Protonet Cao et al. (2019) uses principal component analysis to consider only the resulting leading $d = 60$ dimensions as inputs while zeroing out the rest and the Mixed Protonet is a standard prototypical network trained with a randomized number of shots in the range $[1, 5]$.

We demonstrate the error classification percentage (i.e) 100*p where p is the probability of error classification , we can see that the error classification percentage only increases with the number of shots increasing , both in the training and in the testing phase.$6.(1+\frac{1}{k})(R+R_i)-L\,(||\phi_\theta(x_q)||_2-R_i)^2$ is inversely proportional to $\frac{1}{k}$ for a fixed embedding and a query sample, hence it increases with the number of shots , also the Mixed - shot classification percentage id higher due to the heterogeneous nature of the data, and means more information regarding the distribution of the data.

MODEL CONFIGURATION: Vanilla ProtoNet is used as our baseline . We present the performance of multiple ProtoNets trained with different shots to illustrate the performance degradation issue. ProtoNet-PCA uses principal components of the training split embeddings , with components other than the D leading ones zeroed out. We carry out a parameter sweep on miniImageNet and set d = 60; the same value is used on the other two data sets. For selecting the training shot of the embedding network, we find that overall performance to be optimal using k = 5. we set R = 0.001 N = 85 and randomly choose R_i from a sphere of radius 1 and D = 60 based on performance on miniImageNet

We observe that matching the training shot to the test shot generally provides the best performance for vanilla ProtoNets. Also importantly, training with a mixture of different values of k does not provide optimal performance when evaluated on the same mixture of k values. Instead, the resulting performance is mediocre in all test shots.We obtain the PCA of the embedded data by eigendecomposing the covariance matrix of embeddings, we obtain the principal components expressed as the significant eigenvalues, and the principal directions expressed as the eigenvectors corresponding to those eigenvalues. The number of significant eigenvalues approximates the intrinsic dimension of the embedding space. When the subspace is linear, this approximation is exact; otherwise, it serves as an upper bound to the true intrinsic dimension Fukunaga & Olsen (1971)

6 CONCLUSION AND FUTURE WORK

In this paper we provide a novel bound on the generalization error on the N way k shot classification task using prototypical networks, which is crucial in the sense that existing works hold for large samples of

Model	Training Shots	Testing Shots 1	5
ProtoNet	1	94.07	95.54
ProtoNet	5	97.36	96.38
Mixed Shot	$1-5$	98.65	97.56
PCA ProtoNet	1	94.94	98.85

(a) *Omiglot-20-way*, with 4 layer CNN.

Model	Training Shots	Testing Shots 1	5	10
ProtoNet	1	94.46	99.07	98.35
ProtoNet	5	96.02	98.99	98.19
Mixed Shot	$1-5$	96.53	99.15	98.43
PCA ProtoNet	1	98.45	96.54	96.12

(b) *Omiglot-20-way*, with 7 layer ResNet.

Model	Training Shots	Testing Shots 1	5	10
ProtoNet	1	44.75	64.7	68.90
ProtoNet	5	48.96	68.23	72.23
Mixed Shot	$1-5$	49.36	68.96	72.89
PCA ProtoNet	1	48.36	68.83	65.54

(c) *mini*ImageNet-5-way, with 4 layer CNN.

Model	Training Shots	Testing Shots 1	5	10
ProtoNet	1	52.65	68.27	72.29
ProtoNet	5	47.4	69.63	74.45
Mixed Shot	$1-5$	42.2	68.23	74.54
PCA ProtoNet	1	51.93	69.98	74.8

(d) *mini*ImageNet-5-way, with 7 layer ResNet.

Model	Training Shots	Testing Shots 1	5	10
ProtoNet	1	42.37	62.37	67.98
ProtoNet	5	43.65	63.76	70.45
Mixed Shot	$1-5$	47.78	69.84	75.36
PCA ProtoNet	1	45.65	60.36	72.54

(e) *tiered*ImageNet-5-way, with 4 layer CNN.

Model	Training Shots	Testing Shots 1	5	10
ProtoNet	1	44.64	68.54	72.34
ProtoNet	5	48.65	70.54	76.42
Mixed Shot	$1-5$	55.21	65.98	73.83
PCA ProtoNet	1	51.33	68.88	72.17

(f) *tiered*ImageNet-5-way, with 7 layer ResNet.

Table 1: Error Classification percentage for various Prototypical variants.

		Testing Shots					
Model	Training Shots	1	2	3	4	5	Average Accuracy
Vanilla Proto-Net	1	96.46%	98.39%	98.82%	99.01%	99.07%	$98.35 \pm 0.05\%$
Vanilla Proto-Net	2	95.85%	98.32%	98.80%	98.35%	99.07%	$98.2 \pm 0.05\%$
Vanilla Proto-Net	3	95.35%	98.16%	98.23%	98.36%	99.45%	$98.02 \pm 0.05\%$
Vanilla Proto-Net	4	95.34%	98.99%	96.75%	98.72%	97.49%	$98.22 \pm 0.05\%$
Vanilla Proto-Net	5	96.97%	98.54%	97.69%	98.76%	97.54%	$97.75 \pm 0.05\%$
PCA-Prot-Net	1-5	96.53%	98.73%	98.63%	99.06%	99.15%	$98.43 \pm 0.06\%$
PCA-Proto-Net	1	96.02%	98.22%	98.76%	98.84%	99.03%	$98.19 \pm 0.05\%$

Table 2: Error Classification percentage on Omniglot-20-way, with 7 layer ResNet embedding network.

data and hence cannot be applied to k shot learning , where data samples are limited . We also integrate the prototypical architecture of the network in obtaining the error probability of the task classification hence making it much more accurate for few shot learning. We do not assume homogeneous distribution of samples while classification , making it more relevant to practical applications.Sharpness and accuracy of our bound is also demonstrated on various data sets in the experimental section.

Future work includes obtaining a framework to analyze best possible architecture for k -shot learning specific to the data sets wherein presently , we study classification pertaining to a given architecture , however trying to obtain the best possible architecture mathematically which is better in generalization perspective would

be much more relevant and also efficient way of learning inner working of K shot learning. We would like to work in this direction.

REFERENCES

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Peter Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear vc dimension bounds for piecewise polynomial networks. *Advances in neural information processing systems*, 11, 1998.

Avrim L Blum and Ronald L Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5 (1):117–127, 1992.

Tianshi Cao, Marc Law, and Sanja Fidler. A theoretical analysis of the number of shots in few-shot learning. *arXiv preprint arXiv:1909.11722*, 2019.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pp. 192–204. PMLR, 2015.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, 100(2):176–183, 1971.

Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension bounds for piecewise linear neural networks. In *Conference on learning theory*, pp. 1064–1068. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kenji Kawaguchi. Deep learning without poor local minima. *Advances in neural information processing systems*, 29, 2016.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Husheng Li. Analysis on the nonlinear dynamics of deep neural networks: Topological entropy and chaos. *arXiv preprint arXiv:1804.03987*, 2018.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. *Advances in neural information processing systems*, 27, 2014.

David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 230–234, 1998.

David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pp. 164–170, 1999.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27, 2014.

Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *arXiv preprint arXiv:1905.13344*, 2019.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Felix Otto and Cédric Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 173(2):361–400, 2000.

Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*, 2013.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Alvin C Rencher and G Bruce Schaalje. *Linear models in statistics*. John Wiley & Sons, 2008.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Matus Telgarsky. Benefits of depth in neural networks. In *Conference on learning theory*, pp. 1517–1539. PMLR, 2016.

SS Vallender. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4):784–786, 1974.

Joaquin Vanschoren. Meta-learning. In *Automated machine learning*, pp. 35–61. Springer, Cham, 2019.

Vladimir Vapnik, Esther Levin, and Yann Le Cun. Measuring the vc-dimension of a learning machine. *Neural computation*, 6(5):851–876, 1994.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10 (5):988–999, 1999.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.