Rethought Generalization: Empirical Analysis of Generalization Bounds for Compositionally Sparse Networks

Felix B. Berg
MIT
Cambridge, MA 02139
felixbb@mit.edu

Riccardo Conci
Harvard University
Cambridge, MA 02139
riccardo_conci@fas.harvard.edu

Christian Aagnes
Harvard University
Cambridge MA 02138
christianaagnes@g.harvard.edu

Abstract

Recent advancements in statistical learning theory have introduced tighter generalization bounds for deep neural networks with compositional sparsity, challenging the long-held critique of traditional complexity measures. In this work, we revisit the relationship between theoretical bounds and empirical performance in the context of convolutional neural networks (CNNs). We empirically evaluate tighter generalization bounds focusing on networks trained with varying degree of randomness showing that, while still vacous, they entail great information about the generalization performance of the network. We also investigate how the bound changes with scaled data sizes, indicating with extended data, a non-vacous bound could be achieved. Lastly, we offer insight to how regularization strategies and hyperparameters can be used in future work to generate tight bounds. Our findings underscore the potential of architecture-specific norm-based bounds in generalization capabilities of CNNs.

1 Introduction

Generalization, the ability to successfully learn a function from a subset of labeled data and apply it to novel unseen data from the same distribution, is a central pillar in statistical learning theory [1].

As argued in Zhang et al., [2][3], training accuracy alone, however, is not an adequate proxy for generalisation in deep overparametrised neural networks, as these can easily interpolate random labels even with explicit regularisation. Recent efforts have studied the generalisation performance of deep networks by analyzing the complexity of the learned function, providing a potentially better alternative to upper bound the gap between training and test accuracy.

These bounds, however, fail to take into account the specific architecture of the network. Galanti & Xu [4] overcome this and derive a novel and much tighter bound for compositionally sparse deep networks. These developments provide an opportunity to revisit the critique of Zhang et al., offering potential pathways for reconciling statistical learning theory with modern deep learning.

In this report, we repeat the flavour of experiments set up in Zhang et al, and empirically show the strength of this novel approach in providing tighter bounds on the generalisation error. Moreover, we investigate how this bound varies as a function of dataset size and optimisation parameters.

1.1 Related work

Various methods exist for measuring the complexity of the learned function to provide a bound on the test error. Historically, classical metrics such as Rademacher complexity and VC-dimension failed to give non-vacuous bounds on the generalisation error [5]. This was made evident by the two papers by Zhang et al [2][3].

Recent work has shown tighter generalization guarantees for deep neural networks based on various norms of their weight matrices [6][7] [8] [9]. Truong et al. [10] provide non-vacuous generalization bounds for Convolutional Neural Networks (CNNs) for classifying a small class of images using Rademacher complexity. Moreover, Xiao et al. [11] show that Rademacher complexity remains a valid tool for characterizing generalization performance under adversarial conditions.

Another relevant contribution is the work of Pinto et al. [12], which explores the role of low-rank constraints in limiting the capacity of overparameterized networks. Their approach uses the effective rank of weight matrices as a measure of complexity, showing that networks trained with low-rank regularization can achieve tighter generalization bounds without sacrificing performance.

Although these and many other recent approaches seem interesting, we focus solely on the norm-based generalization bounds proposed by Galanti & Xu, motivated by the framework's emphasis on compositional sparsity, which aligns well with the architecture of the convolutional neural networks used in our experiments.

1.2 Contributions

In this work, we empirically test the bounds for generalization error derived for compositionally sparse networks presented by Galanti & Xu, and analyse their behaviour under varying percentage of random labels and training dataset size. In the former experiment, we discuss the role of explicit regularisation on the generalisation bound and the ability to learn random labels. In the latter experiments, we provide a novel insight on how the generalisation bound scales as a function of dataset size both for true and random labels.

2 Problem setup

2.1 Datasets

We use the MNIST dataset, the standard benchmark used in experiments by Galanti & Xu. MNIST contains a collection of 70,000 grayscale images of handwritten digits, ranging from 0 to 9, with 60,000 images allocated for training and 10,000 for testing. Each image is 28x28 pixels, providing a compact and standardized format for visual pattern recognition tasks.

2.2 Models

For consistency, we keep the CNN architecture used by Galanti & Xu in their experiments: a ConvNet with 3 convolutional layers followed by a fully connected layer. Each convolutional layer has a kernel size of 2, with stride 1 padding 0 and output channels of 200, with weight normalisation between the convolutional layers. This model has a total of 1.6 million parameters, of which 300 thousand come from the convolutional layers and 1.3 million from the fully connected layer.

For optimisation, we use SGD with a learning rate of 0.01 and momentum of 0.9. The batch size is 32 unless explicitly varied for analysis. We set the weight decay to $\lambda = 1\text{e}{-4}$ unless explicitly varied for analysis. This weight decay is much smaller than the original $\lambda = 3\text{e}{-3}$ used in Galanti & Xu. As is discussed below, weight decay has a dose-response effect on the ability to learn random labels. As a result, this decreased weight decay was required for our experiments on random labels.

2.3 Experimental variables

Fraction of random labels: MNIST labels are replaced with uniformly random class assignments to simulate an unstructured learning scenario. We run all experiments for both 0%, 50% and 100% of the training dataset set to random class assignments.

Dataset Size: The size of the training dataset, m, is varied to observe its impact on the generalization bound. Subsets of 10%, 25%, 50%, 75% and 100% of the training data are used to test how the bound scales with increasing values of m.

Regularisers: We further assess the behaviour of the bound on true and random labels as a function of batch size (32, 64 128) and weight decay (0, 0.0001, 0.0005, 0.001, 0.003).

3 Background Theory

The Rademacher complexity, $R_X(F)$ measures the capacity of a hypothesis class F (e.g., neural networks of a certain architecture) to fit random noise. The general empirical Radamacher complexity is given by

$$R_X(\mathcal{F}) := \frac{1}{m}\mathbb{E}_{\xi:\xi_{ir}\sim\mathcal{U}[\pm 1]}\left[\sup_{f_w\in\mathcal{F}}\left|\sum_{i=1}^{m}\sum_{r=1}^{C}\xi_{ir}f_w(x_i)_r\right|\right].$$

This holds for fully connected neural networks and does not take into account properties of the neural network architecture G.

Galanti & Xu provide the following bound on the Radamacher complexity of the class $\mathcal{F}_{G,\rho}$ of networks of architecture G of norm $\leq \rho$, and L layers. We refer you to their paper for details on their specific notation used below.

$$R_X(F_{G,\rho}) \leq \frac{\rho}{m}\cdot\left(1+\sqrt{2\left(\log(2)L+\sum_{l=1}^{L-1}\log(\deg(G)_l)+\log(C)\right)}\right)\cdot \sqrt{\max_{j_0,\ldots,j_L}\prod_{l=1}^{L-1}|\text{pred}(l,j_l)|\cdot\sum_{i=1}^{m}\|z^0_{j_0}(x_i)\|_2^2}.$$

The Radamacher complexity can be related to the prediction error in a probabalistic way given by the following equation:

$$\text{err}_P(f_w)-\text{err}_S^{\gamma}(f_w)\leq\frac{2\sqrt{2}}{\gamma}\cdot R_X(\mathcal{F})+3\sqrt{\frac{\log(2/\delta)}{2m}}.$$

The definition of the the test error is

$$\text{err}_P(f_w)=\mathbb{E}_{(x,y)\sim P}\left[\mathbb{I}\left[\max_{j\neq y}f_w(x_i)_j\geq f_w(x_i)_y\right]\right],$$

and the emprical margin error is then given by

$$\text{err}_S^{\gamma}(f_w)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left[\max_{j\neq y}\left(f_w(x_i)_j\right)+\gamma\geq f_w(x_i)_y\right].$$

From this we see that the distance between the test and training error defined in this way cannot be more than 1.

As noted above, the Rademacher bound does not take into account the architecture of the neural network. This is a major limitation when attempting to create tight bounds. By instead making specific assumptions about compositionally sparse architectures, Galanti & Xu elegantly derive the following bound:

$$\text{err}_P(f_w)-\text{err}_{\gamma}^{S}(f_w)\leq\frac{2\sqrt{2}(\rho(w)+1)}{\gamma m}\cdot\left(1+\sqrt{2\left(\log(2)L+\sum_{l=1}^{L-1}\log(\deg(G)_l)+\log(C)\right)}\right)\cdot \sqrt{\max_{j_0,\ldots,j_L}\prod_{l=1}^{L-1}|\text{pred}(l,j_l)|\cdot\sum_{i=1}^{m}\|z^0_{j_0}(x_i)\|_2^2}+3\sqrt{\frac{\log(2(\rho(w)+2)^2/\delta)}{2m}}.$$

Applying this bound to CNNs we adjust the degree of the nodes in the neural network graph G, $deg(G)$ and the predecessors for each node to k_l, the kernel size squared for each layer l.

$$\text{err}_P(f_w) - \text{err}_\gamma^S(f_w) \leq \frac{2\sqrt{2}(\rho(w)+1)}{\gamma m} \cdot \left(1 + \sqrt{2\left(\log(2)L + \sum_{l=1}^{L-1}\log(k_l) + \log(C)\right)}\right) \cdot \sqrt{\prod_{l=1}^{L-1} k_l \max_{j\in[d_0]} \sum_{i=1}^{m} \|z_j^0(x_i)\|_2^2} + 3\sqrt{\frac{\log(2(\rho(w)+2)^2/\delta)}{2m}}.$$

Here d_0 is the dimension of the input layer, which is $28 \times 28 = 784$ for MNIST. $z_j^0(x_i)$ is the value of the pixel of the input layer of data x_i. The term $\max_{j\in[d_0]} \sum_{i=1}^{m} \|z_j^0(x_i)\|_2^2$ therefore has an upper bound of m.

The $\rho(w)$ term in the bound represents the product of Frobenius norms of the weight matrices across the layers of the network, $\|w\|_F$: $\rho(w) = \prod_{l=1}^{L} \|W^l\|$.

For our architecture we fix the following constants: $\gamma = 1$, $\delta = 0.001$, $L = 4$, $C = 10$, $k_l = 4$. With the upper bound of m we get that

$$\text{err}_P(f_w) - \text{err}_1^S(f_w) \;\leq\; \frac{116(\rho(w)+1)}{\sqrt{m}} \;+\; 3\sqrt{\frac{\log\big(2000(\rho(w)+2)^2\big)}{2\,m}}.$$

The first term is the dominant one.

In the generalization bound for CNNs there is only one scalar that changes during training; $\rho(w)$.

Training on random labels leads to the network memorizing the training set rather than learning a generalizable function. This increased complexity is visible in the increased norms of the weight matrices that define $\rho(w)$. Therefore, we hypothesise that $\rho(w)$ will be smaller for correct labels than random labels and generate a smaller generalization bound.

Also, we see that the choice of m should affect the bound inversely. It is likely, however, that $\rho(w)$ is also a function of m, motivating our expriments on scaling the training dataset size m.

Finally, the final weight matrices are not only dependent on the dataset size and labels, but importantly on the optimisation process. We therefore analyse how the bound is affected by varying the batch size and weight decay.

4 Experiments

In this section we conduct an empirical evaluation of the generalization bound as we vary the percentage of random labels, the size of the dataset, and other optimization parameters such as weight decay and batch size.

Firstly, we note that we managed to reproduce the results of Galanti & Xu with their hyperparameters and architecture and achieve a generalization bound of around 1. However, with their weight decay of 0.003, the model failed to learn random labels. In order to fix this, we decreased the weight decay from 0.003 to 0.0001, which allowed the network to memorize the random labels perfectly. As shown in Figure 1, changing the weight decay has a smoothly varying impact on the generalisation bound of the trained network and on the training accuracy on random labels.

Impact of weight decay

The larger the weight decay parameter, the lower the generalization bound. This is due to the fact that for a fixed dataset size m, the bound only varies with $\rho(w)$ so a higher weight decay results in lower weights and thus also lower norms overall.

The results in Figure 1 provide a more nuanced insight compared to the 'weight decay on or off' experiments discussed in the Zhang et al. Indeed, both with and without weight decay, the models can perfectly interpolate both true and random labels. However, this is only true if the weight decay is low enough for the complexity of the solutions to include memorising random labels.

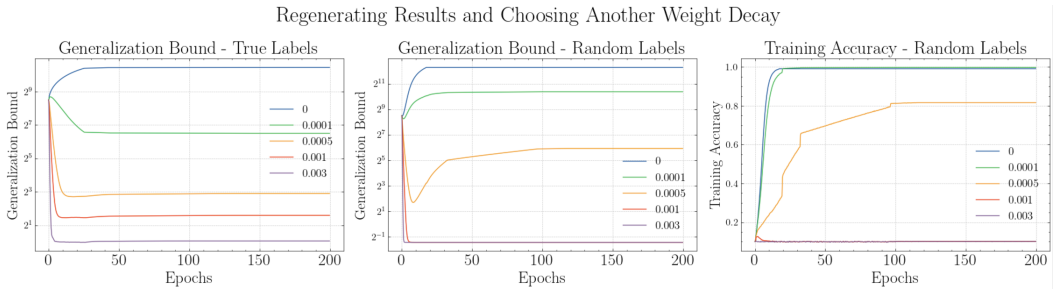

Figure 1: Generalization bound and training accuray for different values of weight decay: 0, 0.0001, 0.0005, 0.001, 0.003.

Interestingly, when the weight decay is too high to memorise random labels (>0.0005), the generalisation bound reaches < 1. In this case, shown in Figure 1, it reaches around 0.3. This bound is unfortunately still vacuous as the training error ≈ 0.1, and in expectation, the test error is upper bounded by the training error. Low generalization bound in itself is not interesting when it is not learning the training set. We will therefore focus on the combination of low bound and good training accuracy.

4.1 Random labels

This initial experiment allowed us to select the lowest weight decay that would perfectly fit random labels and provide a low generalisation bound. The experiments below analyse in more depth how the bound changes with different fractions of random labels.

As shown in Figure 2, the generalization bound is much higher for random labels than for the correct labels, with 50% random labels falling somewhere in between.

We note that the network trained on 50% random labels achieves 50% training accuracy within the first couple of epochs, showing that it very rapidly learns the correct function. This is indeed shown in the high validation accuracy of around 0.97, on the non-random validation set. In this initial period, it treats the false labels as noise. Having learned the correct label pattern, it moves to memorising the rest of the random labels. As it does so, the validation accuracy gradually decreases and the generalisation bound increases.

The 100% random labels initially have a decreasing generalization bound, which we hypothesize is because it is not able to find a pattern, and therefore focuses on weight decay before memorization. It then reaches 100% training accuracy slightly faster than on 50% random labels, potentially because of the complexity of re-training its model weights on these false labels.

Despite learning slightly faster, both random label settings improve their accuracy from 50% to 100% between epochs 25 to 60. This suggests a relationship between memorisation ability and amount of random data examples seen. To strengthen this hypothesis we would need to repeat the experiments with more random label fractions and see if indeed they all follow this pattern.

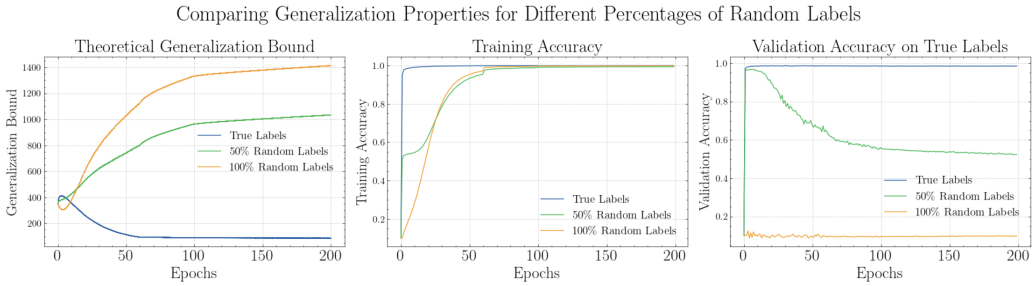

Figure 2: Left: Generalization bound over epochs. Middle: Training accuracy over epochs. Right: Validation accuracy over epochs where the validation set are the true labels, regardless of the randomness during training.

Components of the generalisation bound

$\rho(w)$, the product of the norms of the different layers, is the component of the bound that varies during training. Therefore, we can look in detail at how the norms for each layer vary across training.

As shown in Figure 3, the norms of the convolutional layers decrease during training regardless of labels randomization. Instead the norm of the fully connected layer increases across experiments. The complexity that arises in the random labels over training is mostly manifested in a large increase and deviation in the fully connected layer. As we will see in section 4.3, the layers that are influential for the generalization bound depends on the hyper parameters.

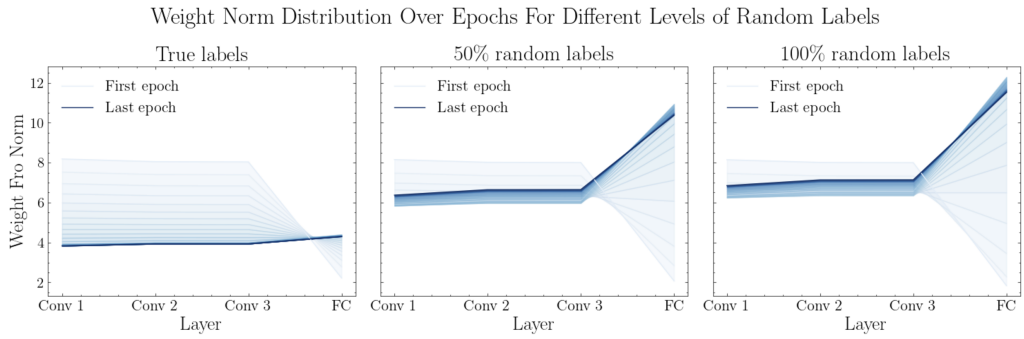

Figure 3: The norm of the layers changing over time across experiments. The opacity shows the variation of the norms across epochs.

4.2 Size of Dataset

As discussed in the background theory section, the bound is approximately proportional to $\rho(w)/\sqrt{m}$. We notice that in the left plot of Figure 4, with increased data size m, the generalization bound will decrease. However, for random labels, more data means more complexity to the network as it has to memorize more data samples. From the right plot in Figure 4 we see that the ρ increases for random labels. The generalization bound therefore decreases more dramatically with the data size for true labels.

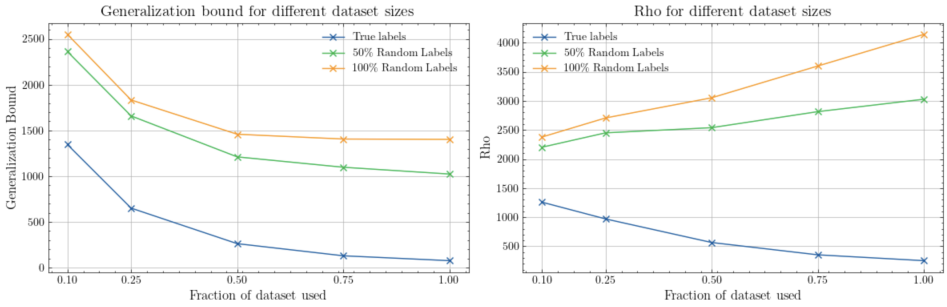

Figure 4: Generalization bound as a function of dataset size m for true, 50% random and 100% random lables.

What would happen if we had access to more data extending m to values larger than 1? In order to discuss this we fit the ρ in the right plot in Figure 4. Removing constants we get approximately the following terms:

$$\rho_{\text{true}}(m) \sim 0.16^m, \quad \rho_{\text{50\% random}}(m) \sim 1 + \frac{1}{2}m, \quad \rho_{\text{100\% random}}(m) \sim 1 + m.$$

Now dividing these by $\sqrt{m}$ we get that the proportionality for the generalization bounds is given by

$$\text{bound}_{\text{true}}(m) \sim \frac{0.16^m}{\sqrt{m}}, \quad \text{bound}_{\text{50\% random}}(m) \sim \frac{1}{\sqrt{m}} + \frac{1}{2}\sqrt{m}, \quad \text{bound}_{\text{100\% random}}(m) \sim \frac{1}{\sqrt{m}} + \sqrt{m}.$$

As we extrapolate these curves beyond the current data fraction of our experiments, we can hypothesise the following three results: for true labels, the bound continues to decrease; for 50% random labels the bound has a minimum at $m = 2$, and for 100% random labels, the extension to $m > 1$ would lead to the bound gradually increasing again. Future work should empirically test this.

4.3 Optimization parameters

How the model achieves low norm weight matrices depends not only on the dataset labels, but importantly on the optimisation process. In the following analyses, we show how the bound is impacted as a function of batch size and weight decay.

Without weight decay, we rely on weight normalization and the implicit regularisation of SGD to learn a function that not only interpolates the training data but also has a strong generalisation performance. We hypothesised that with a smaller batch size the generalisation bound would therefore be smaller due to increased implicit regularisation when learning true labels. As shown on the right-most plot of Figure 5, our results show otherwise, with higher batch size of 128 showing a lower bound. We can look in more detail at the composition of this through Figure 6, and note that the main difference between the norms in batch size 32 and 128 are at the fully connected layer.

When we add a small weight decay on normal labels, this pattern is flipped, with batch size 32 providing the lowest generalisation bound, although all of the batch sizes have lower bounds compared to no weight decay. Looking at the layer-specific norms, we note that batch size 32 leads to lower norms in the convolutional layers and slightly increased in the fully connected, compared to the opposite pattern with batch size 128. The lowest generalisation bound is therefore achieved when we have weight decay and low batch size in combination.

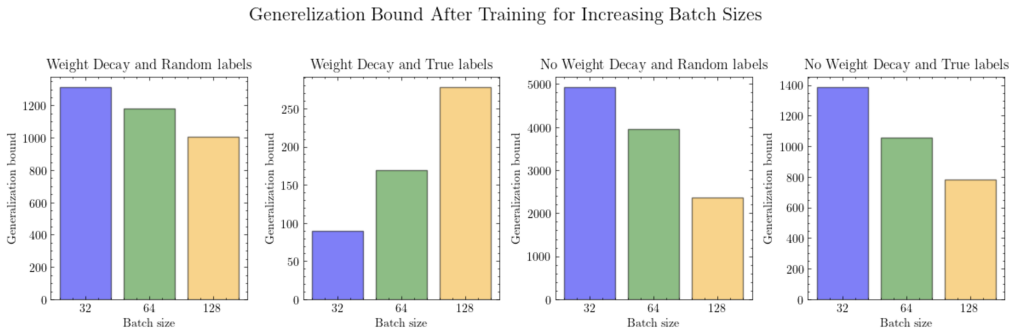

Figure 5: Generalization bound as a function of batch size and weight decay

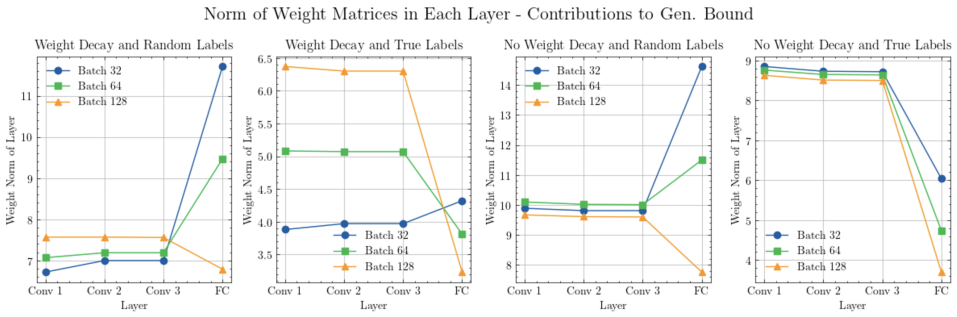

Figure 6: Norms of the different layers for diffenrent batch sizes.

When analysing this for random labels, we would expect the bounds to be much higher in general if the model has successfully memorised the dataset. Across both settings with and without the explicit weight decay regulariser, a bigger batch size led to smaller bounds again driven by the low norms in the fully connected layer.

We notice from Figure 6 that the fully connected layer has the lowest norm for large batch sizes regardless of weight decay and random labels. This is the dominating factor causing subplot 1, 3 and

4 in Figure 5 to have lowest generalization bound for batch size 128. The combination of weight decay and true labels however, manages to get low convolutional norms in all layers.

5 Conclusion

In this paper, we conducted an extensive empirical analysis of the norm-based generalization bound by Galanti & Xu. Our main experiments focused on the following two novelties: 1) evaluating the behaviour of the generalization bound under varying fractions of randomly labeled data and 2) examining how the bound scales with the size of the training dataset. Additionally, we explored the influence of optimization parameters, such as weight decay and batch size, on the bound's tightness and the network's ability to generalize.

Our findings highlight that the generalization bound is significantly higher for networks trained on random labels compared to true labels, with partially randomized labels falling in between. We also observed that the bound for true labels scales approximately as $o(1/\sqrt{m})$ while the random labels scales as $\omega(1/\sqrt{m})$ with the possibility of increasing for larger m. Furthermore, we showed that careful tuning of weight decay and batch size is critical to maintaining low bounds while achieving high training accuracy.

Future work includes testing the bound for larger m to see whether the current extensions are true, or if other interesting behaviour occurs. This can be achieved by scaling the data size using AI generated MNIST data. It would also be interesting to pivot the focus from randomness in output, to randomness in input. Bricken showed that adding noise to input would give rise to sparsity in activation of the neurons [13], and it would be interesting to see how the generalization bound is affected by this.

In conclusion, although the bound is vacous for both random and non-random labels, the bound is a great measure for the quality of generalization. We also see potential in the bound to have the capacity to generate non-vacous bounds for certain parameters and architectures.

References

[1] Ulrike von Luxburg and Bernhard Schoelkopf. Statistical learning theory: Models, concepts, and results. *arXiv [stat.ML]*, October 2008.

[2] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv [cs.LG]*, November 2016.

[3] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, March 2021.

[4] Tomer Galanti, Mengjia Xu, Liane Galanti, and Tomaso Poggio. Norm-based generalization bounds for compositionally sparse neural networks. *arXiv [cs.LG]*, January 2023.

[5] V N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16(2):264–280, January 1971.

[6] P Bartlett and S Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3(Nov):463–482, March 2003.

[7] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401. PMLR, June 2015.

[8] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv [cs.LG]*, December 2017.

[9] Fengxiang He, Tongliang Liu, and Dacheng Tao. Why ResNet works? residuals generalize. *arXiv [stat.ML]*, April 2019.

[10] Lan V. Truong. On rademacher complexity-based generalization bounds for deep learning. *Preprint. Under review*, September 2024.

[11] Jiancong Xiao, Ruoyu Sun, Qi Long, and Weijie J Su. Bridging the gap: Rademacher complexity in robust and standard generalization. *Proc. Mach. Learn. Res.*, 247:5074–5075, 2024.

[12] Andrea Pinto, Akshay Rangamani, and Tomaso Poggio. On generalization bounds for neural networks with low rank layers. *Proc. Mach. Learn. Res.*, XXX:1–16, 2025.

[13] Trenton Bricken, Rylan Schaeffer, Bruno Olshausen, and Gabriel Kreiman. Emergence of sparse representations from noise. *Proc. 40th Int. Conf. Machine Learning*, 202:–, July 2023.