ROBUSTNESS AND/OR REDUNDANCY EMERGE IN OVERPARAMETRIZED DEEP NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

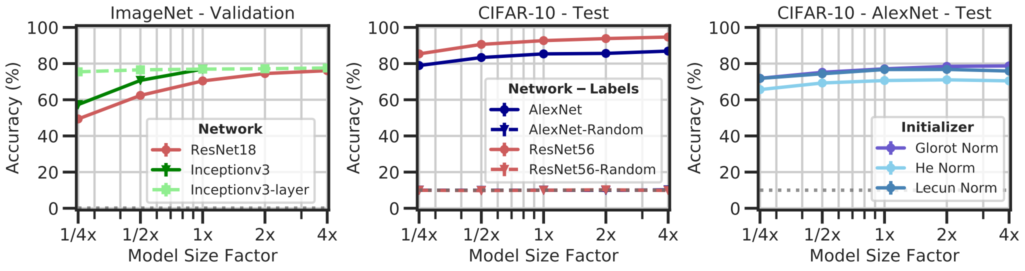
ABSTRACT

Deep neural networks (DNNs) perform well on a variety of tasks despite the fact that most used in practice are vastly overparametrized and even capable of perfectly fitting randomly labeled data. Recent evidence suggests that developing "compressible" representations is key for adjusting the complexity of overparametrized networks to the task at hand and avoiding overfitting (Arora et al., 2018; Zhou et al., 2018). In this paper, we provide new empirical evidence that supports this hypothesis, identifying two independent mechanisms that emerge when the network's width is increased: robustness (having units that can be removed without affecting accuracy) and redundancy (having units with similar activity). In a series of experiments with AlexNet, ResNet and Inception networks in the CIFAR-10 and ImageNet datasets, and also using shallow networks with synthetic data, we show that DNNs consistently increase either their robustness, their redundancy, or both at greater widths for a comprehensive set of hyperparameters. These results suggest that networks in the deep learning regime adjust their effective capacity by developing either robustness or redundancy.

1 INTRODUCTION

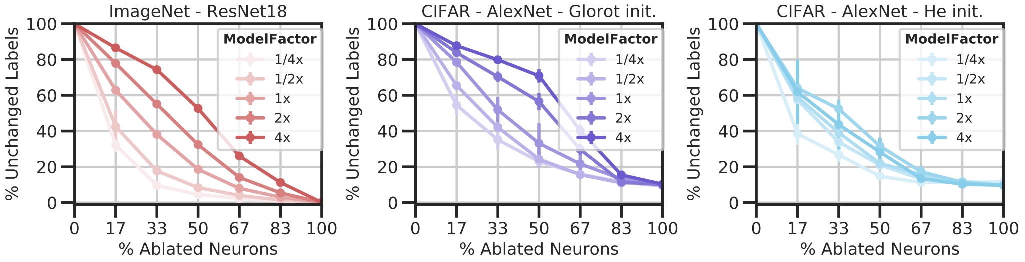
Deep neural networks (DNNs) are capable of successfully learning from examples in a wide variety of tasks. Though these networks are typically trained with large amounts of data, the number of free parameters in their architectures is often several orders of magnitude greater than the number of training examples. This overparametrization reflects the ability of DNNs to memorize entire datasets, even with randomized labels (Zhang et al., 2016; 2017). Additionally, large networks not only tend to match the performance of small ones, but often generalize better (e.g. Neyshabur et al. (2017b); Frankle & Carbin (2018); Neyshabur et al. (2018); Novak et al. (2018)). Figure 1 demonstrates this for a variety of modern networks trained in ImageNet and CIFAR-10. These observations raise the question of how vastly overparametrized networks can perform well in structured tasks without overfitting. While DNNs appear to adapt their capacity to the complexity of the given task, precisely what causes them to do so remains an open question (Poggio et al., 2017).

Several previous studies have aimed to uncover why, out of the many optima an overparametrized network can reach to achieve 100% training accuracy, they tend toward ones that generalize well (Neyshabur et al., 2017b; Zhang et al., 2017; Neyshabur et al., 2018; Novak et al., 2018) often by proving generalization bounds for simple models related to weight matrix norms or Rademacher complexity (Bartlett et al., 2017; Neyshabur et al., 2017a; Arora et al., 2018; Neyshabur et al., 2018). Frankle & Carbin (2018), showed that, in certain networks, the crucial computations were performed by sparse subnetworks within them. In doing so, they suggested that large networks tend to perform as well as or better than smaller ones because they more reliably contained fortuitously-initialized "lottery ticket" subnetworks. Here, we focus on the question of why generalization ability does not decrease as a network's degree of overparametrization increases. We investigate two critical properties of DNNs: robustness (how fragile the network is to removal of units) and redundancy (how similar unit activity is). In doing so, we build off of theoretical work by Arora et al. (2018) and Zhou et al. (2018), connecting the compressibility of DNNs to their non-overfitting behavior. We find that various DNNs train toward regimes with different degrees of robustness and redundancy, but that at least one of the two properties, if not both, consistently emerges as a model's size is increased. Based on these results, we offer interpretations of the various ways in which DNNs may constrain their effective capacity to protect from overfitting.



(a) (b) (c)

Figure 1: **Classification accuracy does not deteriorate with model size factor.** Top-1 accuracies achieved across model sizes and datasets. (a) ResNet18s, Inception-v3s, and Inception-v3s with a single layer varied trained in ImageNet, (b) Regularized AlexNets and ResNet56s with and without training on random labels in CIFAR-10, (c) Glorot, He, and LeCun-initialized AlexNets in CIFAR-10.



(a) (b) (c)

Figure 2: **Some, but not all networks become more robust when model size is increased.** (a) ResNet18 trained in ImageNet, (b) Glorot-initialized AlexNet in CIFAR-10, (c) He-initialized AlexNet in CIFAR-10. Each point shows the proportion of labels that did not change when the percentage of units on the x-axis were randomly ablated. (a) and (b) show an increase in robustness with model width while (b) and (c) illustrate that initialization affects robustness trends.

2 THE ROBUSTNESS-REDUNDANCY HYPOTHESIS

It has been observed that the increase of overparametrization without loss of generality for DNN models relates to adjustments of the models complexity, given the task at hand (e.g. Arora et al. (2018); Zhou et al. (2018)). Observations such as Figure 2a,b show empirically that certain large networks are more robust to the ablation (dropout) of a fixed proportion of units than small ones which might suggest that these large models are adjusting their complexity to the task at hand. However, Figure 2c shows that this robustness can be dependent on network initialization, and to our knowledge, the mechanisms responsible for these observations have not yet been the subject of empirical investigation. Given an increased level of overparametrization, we contemplate six capacity-constraining features which could prevent a network from overfitting. (We cannot a priori exclude the interplay of some of these aspects.)

(i) *Redundant functional units:* units whose activations can be expressed linearly in terms of other units and which affect the output of the network.

(ii) *Redundant blocked units:* units as above but which do not affect output.

(iii) *Nonredundant blocked units:* units with noncorrelated activity which do not propagate to affect output.

(iv) *Silent units:* units that are sparsely activated across datapoints.

(v) *Constant units:* units that are consistently activated with low variance across datapoints.

(vi) *Semantically redundant units:* units which are not linearly correlated to others and which provide a different representation from the previous one, similarly related to the learning task.

Metrics: In this work, we provide measures to aid in distinguishing which of these features that large models may be developing in order to constrain their capacity. Consider a layer ` with n units inside a DNN and a dataset $D = f\mathbf{x}_i; y_i g_{i=1}^m$, with m samples. We define *robustness* as the ability of the network to output consistent labels unperturbed and when unital ablations (dropouts) are applied to a certain proportion of its units. We quantify robustness by applying ablations to units during evaluation as done by (Cheney et al., 2017). We randomly apply these ablations to different proportions of units in our networks to obtain that network's ablation curve as reported in Figure 2. We use the area under these ablation curves as a metric for robustness. This measure should aid in distinguishing features (i, vi) where we expect robustness to increase as n grows larger from features (ii, iii, iv) where we expect it to remain unchanged.

We define *redundancy* as a property of a model which can be quantified through two metrics, compressibility and similarity. Given a layer ` we measure its compressibility by quantifying the amount of principal components needed to explain 95% of the activational variance in the n units. We also introduce *similarity* as a measure of a special type of redundancy in which units are highly correlated. We determine two units in the same layer to be similar if their absolute valued Pearson correlation coefficient exceeds a certain threshold (here, we use 0.5). Figure B2 shows the distributions of these coefficients for several AlexNet models. The use of both compressibility and similarity should discriminate features (i, ii), for which we expect an increase of both measures as n increases, from features (iv, v) for which we expect increasing compressibility and stable similarity (because Pearson correlation normalizes variance).

Redundancy and robustness do not imply each other. To show this, let S and L denote respectively a small and large deep network, both trained on the dataset D. Suppose that the generalization ability of L is equal to or greater than that of S. We provide an example for each of four possible scenarios.

L is more robust and redundant than S: This scenario could be constructed using redundant functional neurons (i). Given S, we could duplicate each layer ` with outgoing weights $w$ in $S$, except for the output layer, so that the new layers were [`; `] composites with $[\frac{w}{2}; \frac{w}{2}]$ outgoing weights. This would increase redundancy and robustness because each unit would have a redundant twin, increasing robustness because each a twin could continue to serve a units function if it were ablated.

L is equally or less robust and more redundant than S: This scenario could result from a "weight balancing" effect. As in the previous case, we could duplicate the neurons in each layer, but also add a large, opposite-signed constant to the halved output weights of each unit and its duplicate so that the new layers were [`; `] composites with $[\frac{w}{2} + \; ; \frac{w}{2} \quad]$ outgoing weights. As grows larger, outputs become less robust to ablation. Such an L would be more redundant, as each unit has a corresponding redundant twin (i), but at the same time less robust as the pairs need to balance each other for stability. A second way this scenario could result is from the addition of silent units (iv) which would tend make activation vectors more compressible (though not more similar), but would not make the network more robust to the ablation of a fixed proportion of units.

L is more robust and equally or less redundant than S: This can occur in certain cases in which the two models learn qualitatively different representations by forming units that are merely semantically redundant (vi). As an example, suppose that S learned holistic class representations but that L had the capacity to learn more complex bag-of-features class representations. If so, L would be more robust because of its bag-like representations without necessarily being more redundant.

L is equally or less robust and redundant than S: Given S, such an L could be created by adding nonredundant blocked units (iii) to S. This operation does not increase redundancy, as the units are not linearly correlated, neither robustness as the smaller proportion of essential units in L would be offset by an increase in the number of units dropped out in with the ablation of a fixed proportion of units.

The robustness-redundancy hypothesis: Consider a small deep network S and a large one L, both with the same architecture, trained with the same explicit regularization scheme, each with a tuned set of hyperparameters and a the same random initialization scheme. Based on our findings, our central hypothesis is that alongside L generalizing as well or better than S, L will be more robust and/or more redundant than S due to the effects of autoregularization.

Dataset	Network	Initialization	Optimizer	Regularizers	L.Rate-B.Size
Uncorr. 10 dim	MLP	Normal?	Momentum	None	Best
Uncorr. 10k dim	MLP	Normal?	Momentum, SGD Adam ?	None	Best
CIFAR-10	AlexNet	Glorot/LeCun/He?	Momentum	None, DA, DO, WD?	Best?
(+ rand. labels?)	ResNet56	Glorot	Momentum	BN, DA, WD	Best?
ImageNet	ResNet18	Glorot	Momentum	BN, DA, WD	Best?
	Inception-v3	Normal	RMSProp	BN, DA, WD	Best

Table 1: **Network training and performance details:** “BN” refers to batch normalization, “DA” refers to data augmentation, “DO” refers to dropout, and “WD” refers to L2 weight decay. “Best” refers to learning rate/batch size combinations that we found to achieve the highest accuracy. Stars (?) indicate factors for which we tested different hyperparameters/variants.

3 Methods

To investigate redundancy and robustness across common cases used for machine learning research, we experiment on a variety of different tasks, networks, initializations, sizes, architectures, optimizers, and regularizers. Table 1 gives details for the training and performance of the networks and datasets we use. Features that we tested multiple variants of are marked with a star (?). Further details for all networks can be found in appendix A. To see how robustness and redundancy vary as functions of a model’s degree of overparametrization, we tested variants of each network in which the number of weights/filters in each layer/block/module were multiplied by factors of 1=4, 1=2, 1, 2, and 4.

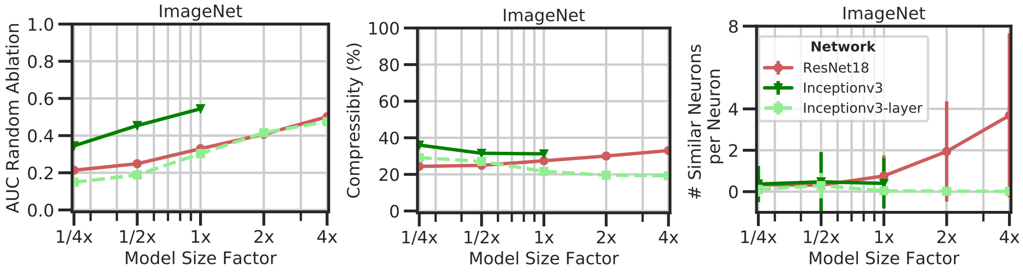
Networks: For experiments with synthetic, uncorrelated data, we used simple multilayer perceptrons (MLPs) with 1 hidden layer of 128 units for the 1x model size and ReLU activations. For experiments using CIFAR-10, we used scaled-down AlexNet models with two convolutional and two fully-connected layers based on Zhang et al. (2016) and ResNet56s with initial convolutions and 3 block layers based on He et al. (2016). For the ImageNet dataset, we used ResNet18s with 4 block layers also based on He et al. (2016) as well as Inception-v3 networks based on (Szegedy et al., 2016). Figure 1 shows the testing accuracy achieved by several of our ImageNet and CIFAR-10 networks. For these and all others we test, increasing model size results in either equal or improved performance. Due to hardware restrictions, we were not able to train any 2x or 4x sized Inception-v3s and instead experimented with versions of these networks where a single layer’s size varied from 1=4x to 4x. Figure A1 plots the number of trainable parameters for each of our networks, showing that they increase exponentially with model size.

Datasets: We used the ImageNet dataset with approximately 1 million training images and 50,000 images for validation and the CIFAR-10 dataset with a 50,000/5,000/10,000 train/validation/test split for our larger-scale experiments. For small-scale ones using MLPs, we used synthetic, uncorrelated data generated by randomly-initialized teacher MLPs with binary output identical in architecture to the 1/4x MLP models that we trained. We verified that our teacher networks output each label for between 40% and 60% of random inputs. We trained and evaluated our MLPs by default on datasets of 1,000 examples. All networks were trained for a fixed number of iterations except for the CIFAR-10 AlexNets which were trained to the epoch of maximum validation accuracy. However, Figure C3 demonstrates that the trends in robustness and redundancy that we analyze are invariant to the amount of training time after convergence in the ResNet18s, and we observe the same for all other models.

Initializations: By default, we initialized our MLP and Inception-v3 networks using random normal distributions with mean 0 and a fixed . In the MLPs, we experimented with various values for . For our AlexNet and ResNet models, we defaulted to using normal Xavier/Glorot initialization with mean 0 and relatively small-variance $\left(\ ^2 = 2=(fan\_in + fan\_out)\right)$. However, in the AlexNets, we also experiment with medium-variance LeCun initialization $\left(\ ^2 = \sqrt{3}=fan\_in\right)$ and high-variance He initialization $\left(\ ^2 = 2=fan\_in\right)$ as well as uniform initialization distributions.

Optimizers: We use RMSProp in the Inception-v3s and the momentum optimizer in all other models by default. We also experiment with using momentumless stochastic gradient descent (SGD) and Adam with our MLPs.

Regularizers: We train our MLPs and AlexNets by default with no explicit regularization (except for early stopping in the AlexNets). We train all ResNets and the Inception-v3 networks with batch



(a) (b) (c)

Figure 3: **Robustness and redundancy emerge in ResNet18s, robustness alone in Inception-v3s (ImageNet):** (a) Areas under ablation curves, (b) principle component compressibility that retains 95% of activational variance, (c) the average number of similar neurons per neuron. Each point is an average across layers.

normalization, data augmentation, and weight decay. To test how explicit regularization contributes to robustness and redundancy, we also analyze AlexNets trained with data augmentation, dropout, weight decay, and all three combined.

Learning Rates and Batch Sizes: Typically in DNNs, varying batch size and learning rate jointly by a constant factor tends to affect training time but not performance (Krizhevsky, 2014), and they are commonly varied in practice. To see how this affects robustness and redundancy, we experiment with varying learning rate and batch size jointly by factors of $1/4$, 1, and 4.

Samplings and Replicates: Due to the the number of units in the models and the size of the datasets, fully analyzing all activation vectors for convolutional filters was intractable. Instead, we based our measures of robustness and redundancy on a sampling of units capped at 50,000 per layer. We average across three independent samplings and find that the variance between them is negligible. For each model, we also average across layers, and except for the ResNet18s and Inception-v3s, we also average results across three independently-trained replicates. All error bars for robustness and compressibility plots show standard deviation between independently trained model replicates when applicable and independent samplings of units when not. They are typically to small to appear in plots. Those for similarity show average standard deviation within trials. For all experiments, we display only results for the test/validation set because results from the train set (except for accuracy) were almost identical.

4 Results

Here, we present our findings for how robustness and redundancy vary as a function of model size. We experiment across a wide variety of networks, hyperparameters, and training methods, show that robustness and/or redundancy emerge in large models across cases used throughout modern machine learning research.

Robustness emerges and redundancy varies in modern ImageNet models. Figure 3 shows that the ResNet18 and Inception-v3 models both become more robust as their model size increases. However, they demonstrate the independence of redundancy and robustness with the ResNet18s developing more compressibility and much more similarity, and the Inception-v3s losing compressibility with size. We take this discrepancy between robustness and redundancy trends as strong evidence that these models, particularly the Inception-v3s, are autoregularizing largely by forming qualitatively different representations at different sizes with units that are merely semantically redundant at large sizes. We also observe that compressibility and similarity do not predict each other particularly well, especially in the case of the ResNet18s, potentially due to different proportions of redundant and silent or constant units in these networks.

Robustness and redundancy develop in networks trained on randomly-labeled data. Our central question revolves around how models are able to constrain their effective capacity to the task at hand, so it is natural to study robustness and redundancy in networks trained to memorize randomly

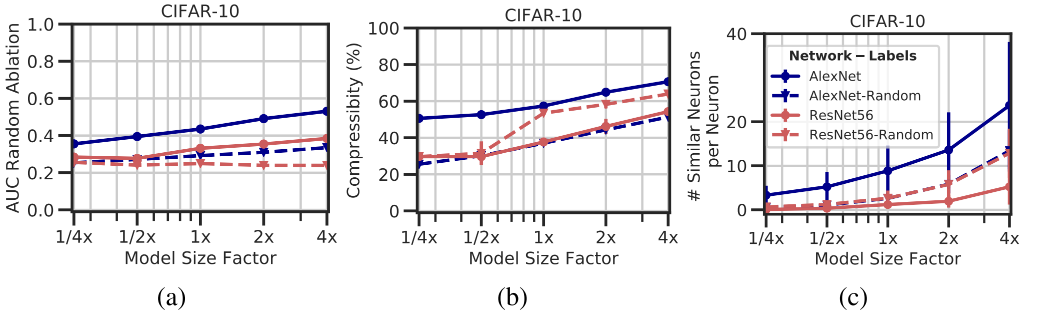


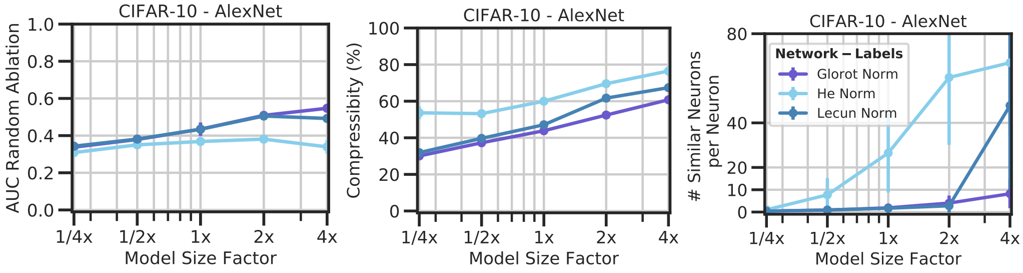
Figure 4: **Robustness and/or redundancy emerge in AlexNet and ResNet18 models, even when trained on randomly labeled data (CIFAR-10):** (a) Areas under ablation curves, (b) principle component compressibility that retains 95% of activational variance, (c) the average number of similar neurons per neuron. Each point is an average across layers.

labeled images. Figure 4 compares our results with the standard AlexNet and ResNet56 models trained on both uncorrupted and randomly-labeled CIFAR-10. With random labels, all networks of the 1x model size or greater were able to fit the training set with perfect or near-perfect performance. Even when fitting randomly labeled data, these models develop significant levels of redundancy and/or robustness at large sizes. Curiously, in the ResNet56 models, fitting random labels caused robustness to flatline but redundancy to *increase* relative to the models trained on correctly-labeled data. These trends strongly indicate that robustness and redundancy do not predict generalization capability.

Robustness and redundancy are sensitive to initialization variance. Some recent work has suggested that network initialization has a large influence over generalization behavior (Frankle & Carbin, 2018; Chizat et al., 2019; Woodworth et al., 2019). To see what effects it may have on robustness and redundancy, we test initializations with differing levels of variance. Figure 5 presents results for AlexNets trained with high-variance He, medium-variance LeCun, and low-variance Glorot initializations. Robustness increases with model size for the Glorot and Lecun-initialized nets but exhibits a fairly flat trend for the He-initialized nets, which demonstrate a case where robustness does not emerge at large size. For the Lecun and He-initialized nets, some model size factor doublings coincide with more than a doubling of the number of similar units per unit, perhaps reflecting a fairly sharp transition to a differently-behaving regime when initialization variance becomes high enough. That fact that the He-initialized nets exhibit a strongly positive trend in redundancy but not robustness might be indicative of redundant blocked units. We also test AlexNets with uniform He, LeCun, and Glorot initializations and find their results (Figure C4) to be very similar to those of the normal initialized ones, suggesting that the initialization distribution matters little compared to the initialization variance.

Adding to our analysis of initializations, Figure C5 shows the results of altering initialization variance in our MLPs trained on uncorrelated data. Those initialized with small variance develop more redundancy at larger model sizes, the opposite trend as in our AlexNets. We also find that the MLPs initialized with very high variance develop neither more redundancy nor robustness. However, Figure C6 shows that the same is not the case for similar MLPs trained on low dimensional data, suggesting that this phenomenon in which neither robustness or redundancy increase is related to high dimensionality in data. We restrict our robustness-redundancy hypothesis only to deep models and show that it applies to state of the art networks, but this case with a single layer MLP, high initialization, and high dimensional, uncorrelated data presents a limitation for our framework. We speculate that these models may be operating in a regime with unique representations or with nonredundant blocked units at large size factors.

The robustness-redundancy hypothesis holds under a number of additional tests. Figure C7, Figure C8, Figure C9, and Figure C10 show that while individual layers in our ResNets and Inception-v3s display unique trends, each develops more redundancy or robustness at higher sizes and generally follows the trend of its network as a whole. There is no consistent relationship between a layer's depth and its robustness or redundancy.



(a) (b) (c)

Figure 5: **Initialization influences robustness and redundancy; high variance initialization leads to redundancy without robustness in AlexNets (CIFAR-10)** (a) Areas under ablation curves, (b) principle component compressibility that retains 95% of activational variance, (c) the average number of similar neurons per neuron. Each point is an average across layers.

Because we speculate that redundancy and robustness in DNNs is a result of autoregularization, it is also natural to analyze their trends in networks trained with and without explicit regularization. Figure C11 shows that data augmentation, dropout, weight decay, and all three together change the overall amount of redundancy or robustness developed in the CIFAR-10 AlexNet models, but they did not change the trends.

To probe the influence of optimizers, we test training the MLP models using stochastic gradient descent (SGD), momentum, and Adam. Figure C12 shows that varying these optimizers can affect how much redundancy or robustness develop, but they do not change overall trends. Additionally, to see if robustness and redundancy depend on learning rate and batch size, we test the ResNet18, ResNet56, and AlexNet models trained on ImageNet and CIFAR-10 with several experiments in which we vary learning rate and batch size while keeping them proportional. Figure C13 and Figure C14 show that this has little to no effect on outcomes.

5 Related Work and Discussion

In this work, we empirically analyze models in terms of their activations (Novak et al., 2018; Morcos et al., 2018a;b) which makes our results contextual to input data. Because of this, we are able to scale our analysis to state of the art networks like ResNet18 and Inception-v3. And by focusing not on the broad question of generalization, but on the subproblem of why networks do not perform worse when their size is increased, we are able to show that redundancy and robustness are central to how networks autoregularize.

A related branch of work has focused on the relationship between a network's compressibility and its generalization behavior (Zhou et al., 2018), and insights by Alvarez & Salzmann (2016); Arpit et al. (2017); Maennel et al. (2018); Ansuini et al. (2019), and Gidel et al. (2019) suggest that DNNs preferably learn simple representations over complex ones. Network compression approaches based on pruning unimportant units or weights developed by Han et al. (2015); Raghu et al. (2016); Hu et al. (2016); Advani & Saxe (2017); Frankle & Carbin (2018) and Li et al. (2018) have suggested that the crucial computations within networks only tend to be performed by a relatively small subset of the weights and units within. Meanwhile, redundancy-based compression methods which prune or merge redundant units have been developed by others including Gong et al. (2014); Srinivas & Babu (2015), and Wang et al. (2019). Our results generally validate both of these approaches, but we show that different networks develop different compressible features and to different extents, so we speculate that both pruning unimportant units and compressing redundant units may be complementary tools for developing new compression algorithms. We also show that redundancy is highly sensitive to a network's initialization while its accuracy is not. This suggests that certain compression techniques could be improved greatly by validating over multiple initializations in order to produce maximally redundant models. We also make progress toward tightening our understanding of how compressible DNNs are which Zhou et al. (2018) shows can lead to improved practical generalization bounds.

Arora et al. (2014) suggests that redundancy implies robustness, and Morcos et al. (2018b) connects a network's robustness to the flattening of a layers' activation space along the direction of a single activation vector to improved generalization performance. However, our findings suggest that these trends may not hold for all networks and that redundancy and robustness poorly predict generalization. Our work is also related to Maennel et al. (2018) who takes a theoretical approach to show that model networks in the overparametrized regime tend to develop weight vectors that align to a set of discrete directions that are determined by the input data. Our work suggest that their conclusions may retain a high degree of explanatory power in some but not all state of the art cases.

Despite a great deal of recent progress, to our knowledge, ours is the first work to date that has quantitatively studied the connections between overparametrization, robustness, and redundancy together. We analyze these phenomena across a wide range of networks which may aid in understanding how well theoretical findings (which are typically based on simple models) generalize to common networks in machine learning. We find that each network we analyze displays unique trends in robustness, compressibility, and similarity, yet that all deep ones develop more redundancy and/or robustness at large model sizes. We also demonstrate that the two are highly dependent on initializations and that high variance increases redundancy in some networks and decrease it in others. Limitations of our work include that we do not analyze cases with varying network depth and the fact that our single-layer MLPs with large initializations trained with high-dimensional, uncorrelated data do not seem to develop either increased robustness or redundancy at large model sizes. However, a recent strand of research has emerged illluminating similarities between deep networks and kernel machines (Belkin et al., 2018; Jacot et al., 2018; Liang & Rakhlin, 2018) and suggesting that networks with high-variance initializations can operate in a kernel-like regime (Chizat et al., 2019; Woodworth et al., 2019) which we suspect relates to these findings for networks initialized with large variance.

6 Conclusion

In this paper, we jointly analyze the robustness and redundancy of deep neural networks with the aim of understanding why generalization ability does not tend to decrease as a network's degree of overparametrization increases. In doing so, we find that robustness and redundancy do not imply each other but that one or the other or both consistently increase alongside overparametrization. We connect these observations to various capacity-constraining features which DNNs may develop in order to support the connection between compressibility and generalization and to shed light on the features networks may develop to avoid overfitting. In doing so, we paint a more complex picture of robustness and redundancy than much previous work has assumed. By illustrating the relationships between these phenomena, we suggest various new research directions in theory of learning and compression. We believe that together, these findings represent a milestone in understanding the emergent properties of overparametrized neural networks.

References

Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, pp. 2270–2278, 2016.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *arXiv preprint arXiv:1905.12784*, 2019.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *International Conference on Machine Learning*, pp. 584–592, 2014.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*, 2018.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 233–242. JMLR. org, 2017.

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

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. *arXiv preprint arXiv:1802.01396*, 2018.

Nicholas Cheney, Martin Schrimpf, and Gabriel Kreiman. On the Robustness of Convolutional Neural Networks to Internal Architecture and Weight Perturbations. *arXiv preprint*, mar 2017. URL http://arxiv.org/abs/1703.08245.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. 2019.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in deep linear neural networks. *arXiv preprint arXiv:1904.13262*, 2019.

Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pp. 1135–1143, 2015.

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.

Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the Intrinsic Dimension of Objective Landscapes. In *International Conference on Learning Representations (ICLR)*, apr 2018. URL http://arxiv.org/abs/1804.08838.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel" ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387*, 2018.

Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes relu network features. *arXiv preprint arXiv:1803.08367*, 2018.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, pp. 5727–5736, 2018a.

Ari S. Morcos, David G. T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *International Conference on Learning Representations (ICLR)*, mar 2018b. URL http://arxiv.org/abs/1803.06959.

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

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pp. 5947–5956, 2017b.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. 2018.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and Generalization in Neural Networks: an Empirical Study. In *International Conference on Learning Representations (ICLR)*, 2018.

Tomaso Poggio, Kenji Kawaguchi, Qianli Liao, Brando Miranda, Lorenzo Rosasco, Xavier Boix, Jack Hidary, and Hrushikesh Mhaskar. Theory of deep learning iii: explaining the non-overfitting puzzle. *arXiv preprint arXiv:1801.00173*, 2017.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. *arXiv preprint arXiv:1606.05336*, 2016.

Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Wenxiao Wang, Cong Fu, Jishun Guo, Deng Cai, and Xiaofei He. Cop: Customized deep model compression via regularized correlation-based filter-level pruning. *arXiv preprint arXiv:1906.10337*, 2019.

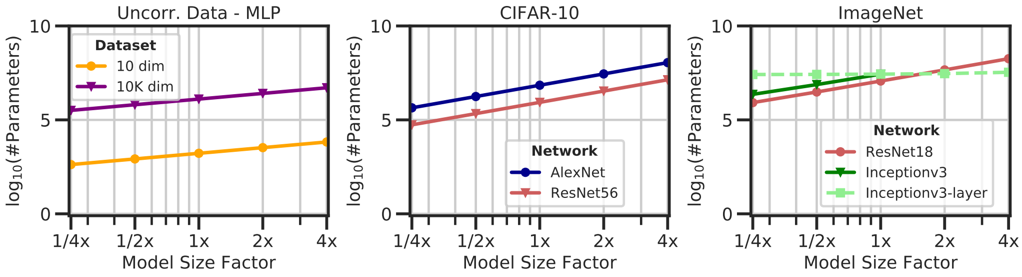
Blake Woodworth, Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Kernel and deep regimes in overparametrized models. *arXiv preprint arXiv:1906.05827*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Chiyuan Zhang, Qianli Liao, Alexander Rakhlin, Karthik Sridharan, Brando Miranda, Noah Golowich, and Tomaso Poggio. Theory of Deep Learning III: Generalization Properties of SGD. *CBMM Memo*, (067), 2017.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. *arXiv preprint arXiv:1804.05862*, 2018.

A Network Details



(a) (b) (c)

Figure A1: **Parameters:** (a) Multilayer perceptrons, (b) AlexNets and ResNet56s, (c) ResNet18s, Inception-v3s, and Inception-v3s with a single layer varied. The log number of trainable parameters at each model size.

ResNet18s: These networks were off the shelf from He et al. (2016) for the ImageNet dataset. They consisted of an initial convolution and batch norm followed by 4 building block (v1) layers, each with 2 blocks and a fully connected layer leading to a softmax output. All kernel sizes in the initial layers and block layers were of size $7 \quad 7$ and stride 2. All activations were ReLU. In the 1x sized model, the convolutions in the initial and block layers used 64, 64, 128, and 256 filters respectively. After Xavier/Glorot initialization, we trained them for 90 epochs with a default batch size of 256 an initial default learning rate of 1 which decayed by a factor of 10 at epochs 30, 60, and 80. Training was done on the ILSVRC 2012 dataset with approximately 1 million images, and evaluation was done on 50,000 validation images. Optimization was done with SGD using 0.9 momentum. We used batch normalization, data augmentation with random cropping and flipping, and 0.0001 weight decay.

Inception-v3s: These networks were off the shelf from Szegedy et al. (2016) for the ImageNet dataset. For the sake of brevity, we will omit including the architectural details here. After using a truncated normal initialization with $\quad = 0{:}1$, we trained them for 90 epochs with a default batch size of 256 and initial default learning rate of 1 with an exponential decay of 4% every 8 epochs. Training was done for 90 epochs on the ILSVRC 2012 dataset with approximately 1 million images, and evaluation was done on 50,000 validation images. Optimization was done with the RMSProp optimizer. We used a weight decay of 0.00004, augmentation with random cropping and flipping, and batch norm with 0.9997 decay on the mean and an epsilon of 0.001 to avoid dividing by zero. Due to hardware constraints, we were not able to train 2x and 4x variants of the network. Instead, we trained the 1/4x-1x sizes along with versions of the network with 1/4x-4x sizes for a the "mixed 2: 35 x 35 x 288" layer only.

AlexNets: We use a scaled-down version of the network developed by Krizhevsky et al. (2012) for the CIFAR10 dataset similar to the one used by Zhang et al. (2016). The network consisted of a 5-layer neural network with two convolutional layers, two dense layers and a readout layer. In each convolutional layer, $5 \quad 5$ filters with stride 1 were applied, followed by max-pooling with a $3 \quad 3$ kernel and stride 2. Importantly, local response normalization with a radius of 2, alpha $= 2 \quad 10^{-05}$, beta $= 0{:}75$ and bias $= 1{:}0$ was applied after each pooling which has an effect of negatively correlating different units. Each layer contained bias terms, and all activations were ReLU. In the 1x sized model, the convolutions used 96 and 256 filters, while the dense layers used 384, and 192 units. We trained these networks on 45,000 images with early stopping based on maximum performance on a 5,000 image validation set. The test set was 10,000 images. Weights were optimized with SGD using 0.9 momentum with an initial learning rate of 0.01, exponentially decaying by 5% every epoch. By default, and unless otherwise stated, we used Xavier/Glorot initialization, a batch size of 128, and no explicit regularizers.

ResNet56s: These networks were off the shelf from He et al. (2016) for the CIFAR-10 dataset. They consisted of an initial convolution and batch norm followed by 3 building block (v1) layers, each with 9 blocks, and a fully connected layer leading to a softmax output. Kernels in the initial layers and block layers were of size $3 \quad 3$ and stride 1. All activations were ReLU. In the 1x sized model, the convolutions in the initial and block layers used 16, 16, 32, 64, and 128 filters respectively. After initializing with Xavier/Glorot initialization, we trained them on 45,000 images for 182 epochs with a default batch size of 128 and an initial default learning rate of 1 which decayed by a factor of 10 at epochs 91 and 136. Testing was done on 10,000 images. Optimization was done with SGD using 0.9 momentum. We used batch normalization, data augmentation with random cropping and flipping (except for our variants trained on randomly labeled data), and 0.0002 weight decay.

MLPs: We use simple multilayer perceptrons with either 10 or 10,000 inputs and binary output. They contained a single hidden layer with 128 units for the 1x model sizes and a bias unit. All hidden units were ReLU activated. Weights were initialized using a normal distribution with default of 0.01. Each was trained by default for 50 epochs on 1,000 examples produced by a 1/4x sized teacher network with the same architecture which was verified to produce each output for between 40% and 60% of random inputs.

B Distribution of Correlation Coefficients

Our criterion for similarity was based on the Pearson correlation r between two units in a layer. We considered two units to be similar if $abs(r)$ was at least 0.5. Figure B2 shows the distribution of all absolute valued correlation coefficients in unregularized, regularized, and random-label-fitting AlexNets in CIFAR-10. In each of these networks, more similar neurons are found in the first convolutional and final fully connected layers, and with the exception of the fully connected layers in the regularized models, the tails of the distributions extend higher at higher model sizes.

C Supplemental Figures

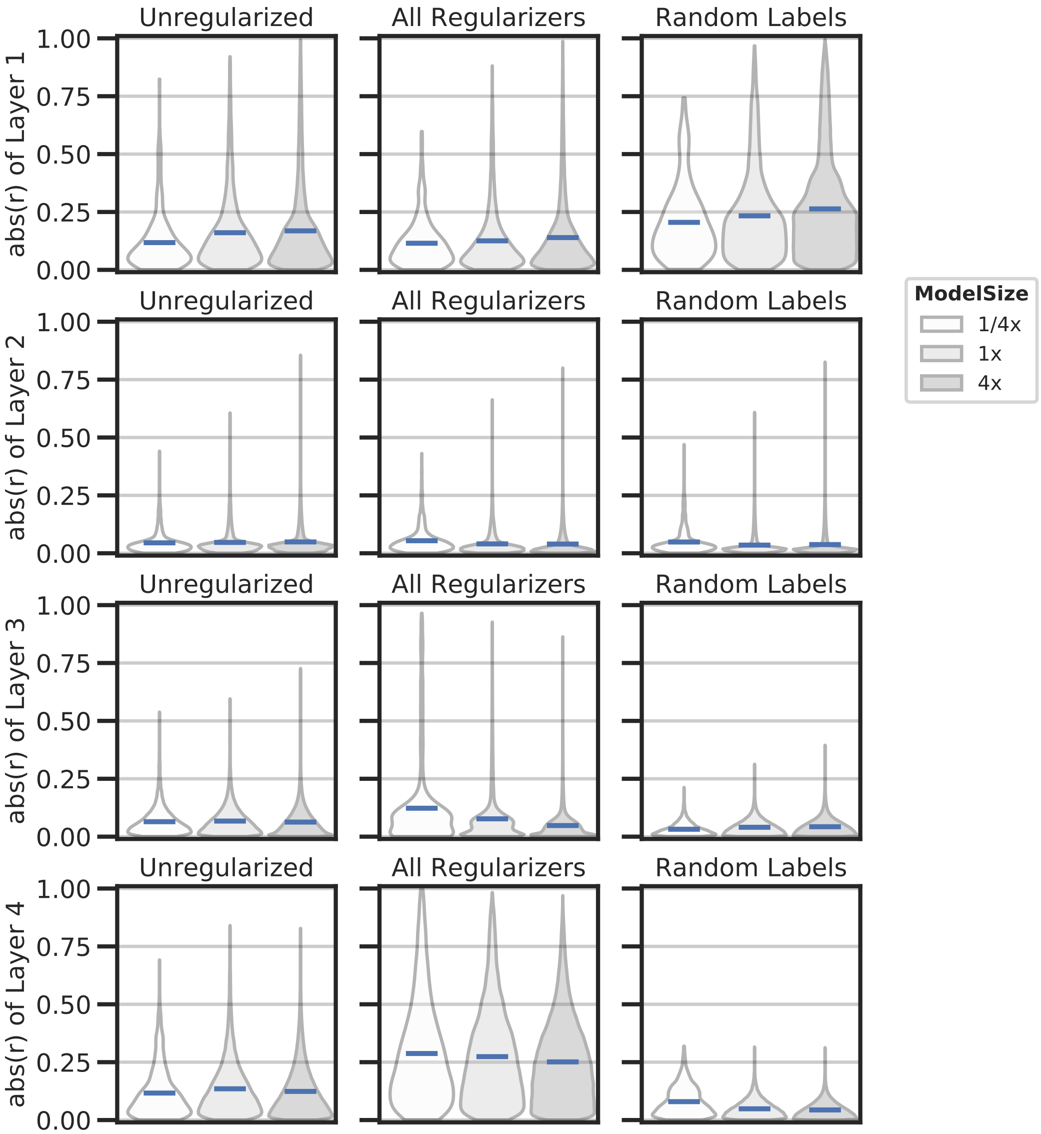
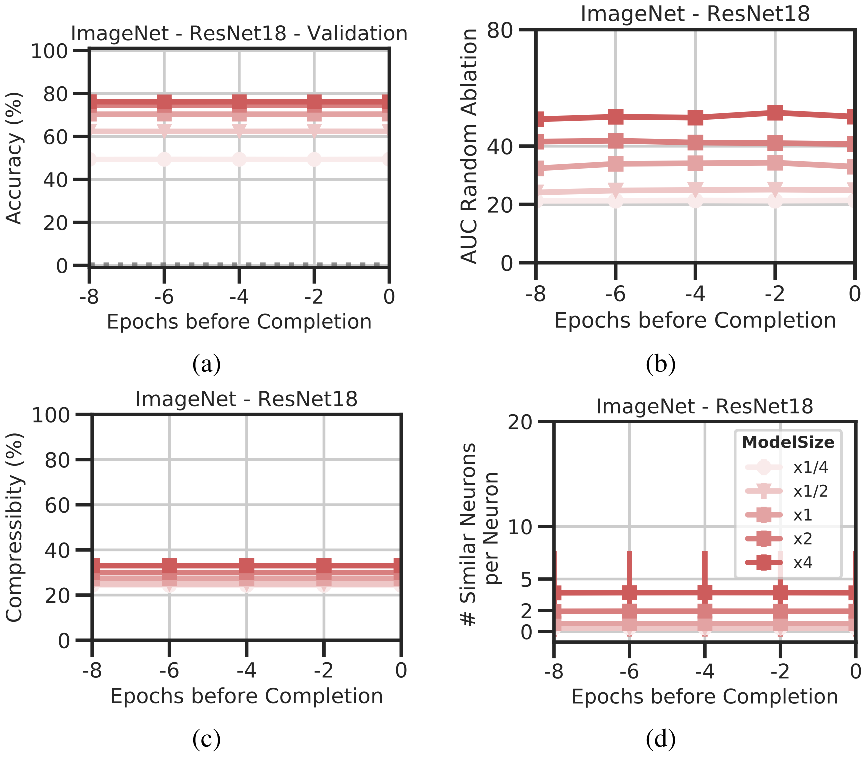


Figure B2: **Absolute value correlation coefficients between neurons in AlexNet models:** The columns correspond to unregularized, regularized (data augmentation, dropout, and weight decay), and random-label-fitting AlexNets in CIFAR-10. The rows correspond to hidden layers of the networks. The first two of which are convolutional and the last two fully connected. Blue bars give means. Each plot shows the distribution for the 1/4x, 1x, and 4x model sizes.



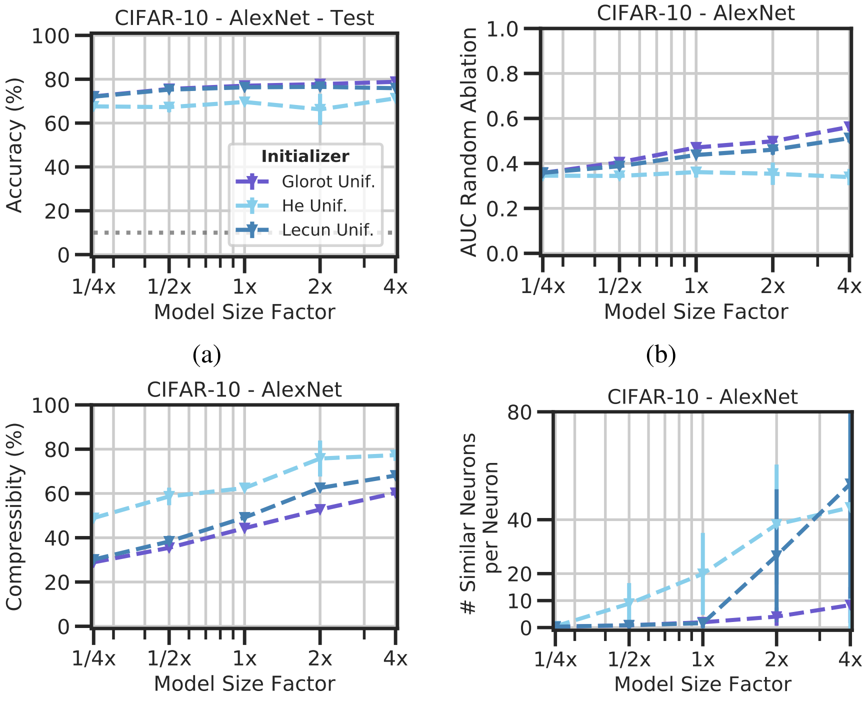
(a)

(b)

(c)

(d)

Figure C3: **Robustness and redundancy are invariant in ResNet18s under different amounts of training past convergence (ImageNet).** Trends in (a) accuracy, (b) robustness, (c) compressibility, and (d) similarity across training epochs after convergence in ResNet18.



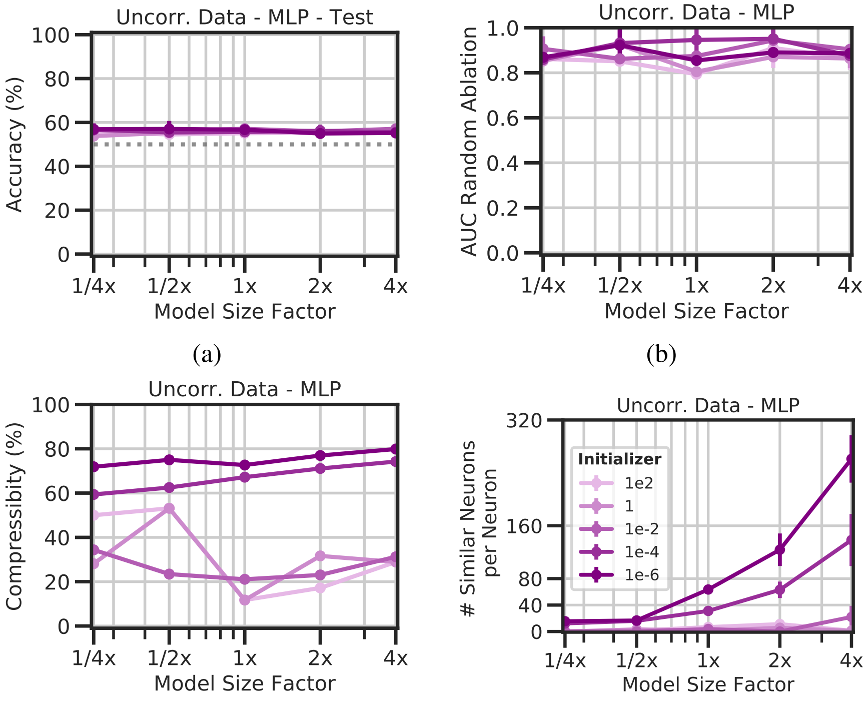
(a)

(b)

(c)

(d)

Figure C4: **AlexNet robustness and redundancy trends with uniform initializations resemble those with normal initializations (CIFAR-10).** Trends in (a) accuracy, (b) robustness, (c) compressibility, and (d) similarity with He, LeCun, and Glorot uniform initializations across size factors for AlexNets.



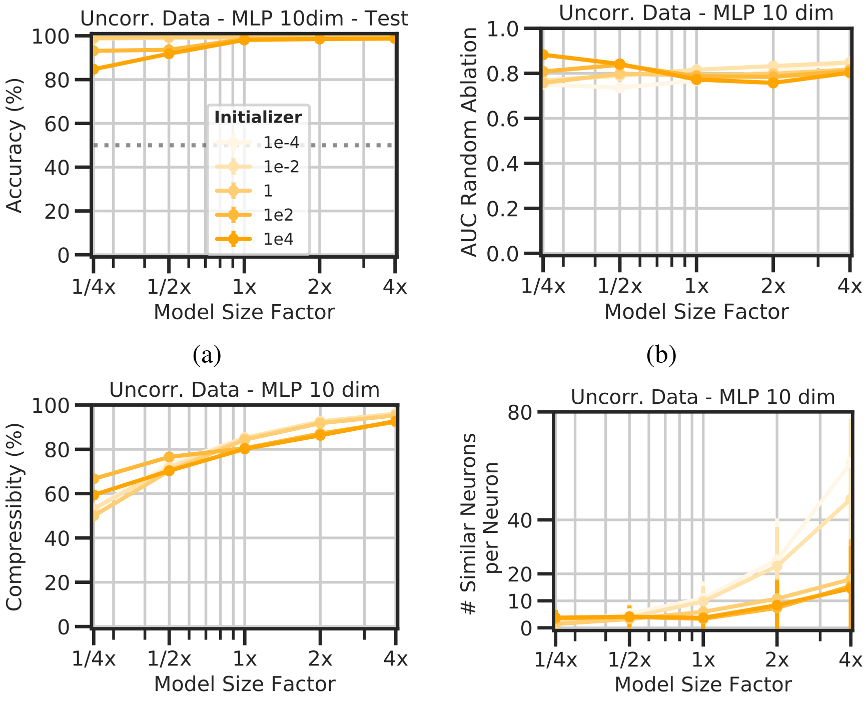
(a)

(b)

(c)

(d)

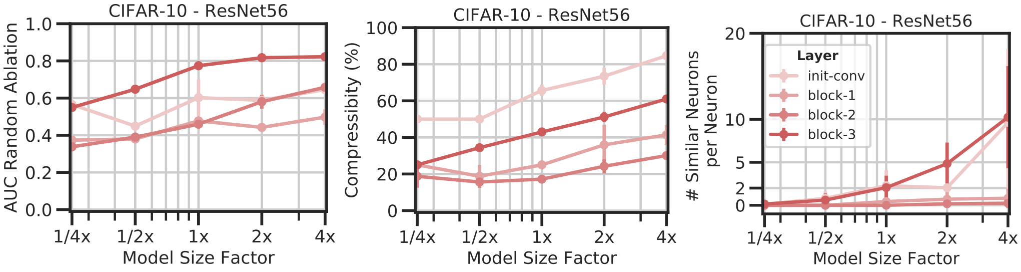
Figure C5: **The emergence of redundancy is sensitive to initialization variance in MLPs trained on 10,000 dimensional data.** Trends in (a) accuracy, (b) robustness, (c) compressibility, and (d) similarity with multiple initialization variances across size factors for MLPs trained on 10,000 dimensional synthetic uncorrelated data. The legend gives for the normal weight initialization. Each initialization results in similar, flat accuracy and robustness curves. Low variance initializations developed more compressibility and similarity with size, but high variance did not.



(a) (b)

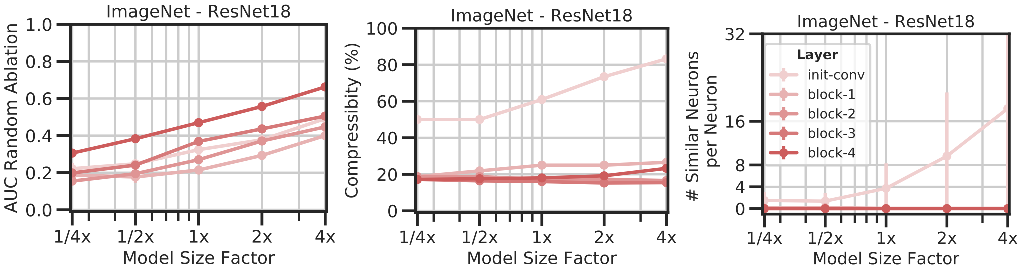
(c) (d)

Figure C6: **The emergence of redundancy is not sensitive to initialization variance in MLPs trained on 10 dimensional data.** Trends in (a) accuracy, (b) robustness, (c) compressibility, and (d) similarity with multiple initialization variances across size factors for MLPs trained on 10,000 dimensional synthetic uncorrelated data. The legend gives for the normal weight initialization. Each initialization results in similar, flat accuracy and robustness curves. All models developed more compressibility and similarity with size.



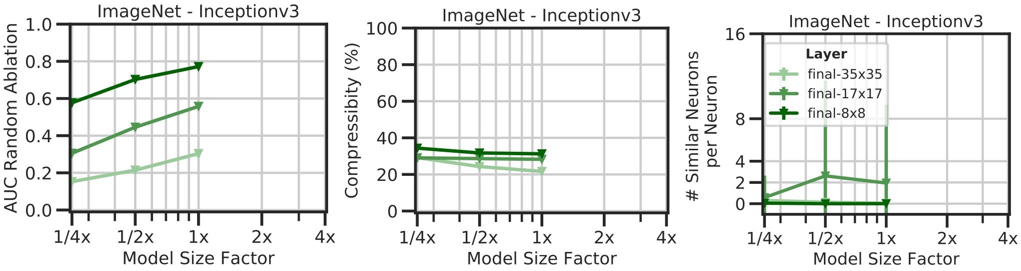
(a) (b) (c)

Figure C7: **ResNet56 robustness and redundancy layerwise (CIFAR-10):** Trends in (a) robustness, (b) compressibility, and (c) similarity among layers/blocks within ResNet56 in CIFAR-10. Each layer/block has a unique curve, with the initial convolutional and final block layers exhibiting the most robustness and redundancy.



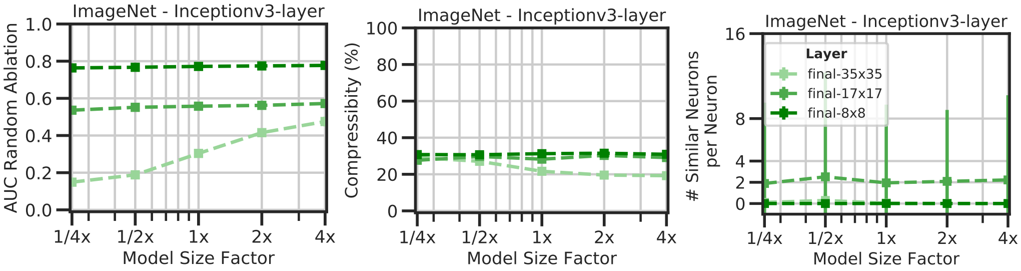
(a) (b) (c)

Figure C8: **ResNet18 robustness and redundancy layerwise (ImageNet):** Trends in (a) robustness, (b) compressibility, and (c) similarity among layers/blocks within ResNet18 in ImageNet. The final block layer is the most robust while the initial convolutional layer is the most compressible and similar. This might be a sign of the phenomenon we describe as "weight balancing" or the formation of silent or constant units in the block layers, particularly the final one.



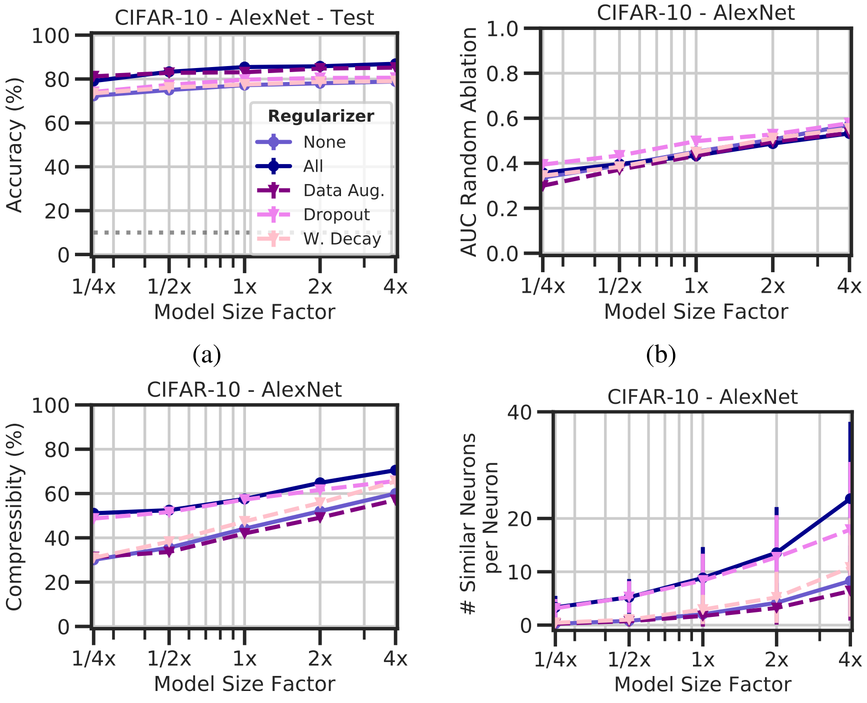
(a) (b) (c)

Figure C9: **Inception-v3 robustness and redundancy layerwise (ImageNet):** Trends in (a) robustness, (b) compressibility, and (c) similarity among the final 35 35, 17 17, and 8 8 blocks within Inception-v3 in ImageNet. Layers develop robustness and compressibility in order of their depth in the network, while the final 17 17 layer is the most similar.



(a) (b) (c)

Figure C10: **Inception-v3 single layer robustness and redundancy layerwise (ImageNet):** Trends in (a) robustness, (b) compressibility, and (c) similarity among the final 35 35, 17 17, and 8 8 blocks within Inception-v3s in ImageNet as only the final 35 35 layer is varied in size. Varying the size of a single layer has little or no effect on the redundancy and robustness of other layers.



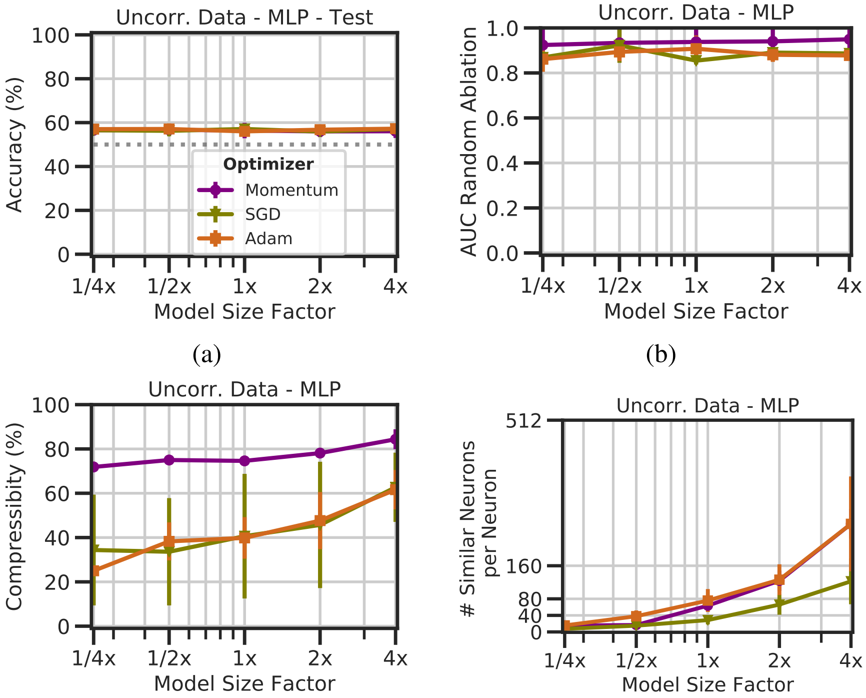
(a)

(b)

(c)

(d)

Figure C11: **Explicit regularization affects amounts but not trends for robustness and redundancy in AlexNets (CIFAR-10).** Trends in (a) accuracy, (b) robustness, (c) compressibility, and (d) similarity with various explicit regularizers across size factors for AlexNets. We test data augmentation, dropout, weight decay, and all three together against the unregularized control. Each regularizer positively affects generalization performance, though weight decay only has a slightly positive effect. None serve to change the general trends in robustness and redundancy, but they scale the curves up or down. Data augmentation reduces robustness, compressibility, and similarity, possibly because it forces the AlexNets to approximate a more complex function than nonaugmented data does. Dropout and weight decay, however, have a large and small positive effect on robustness, compressibility, and redundancy respectively. We attribute this to dropout forcing networks to develop redundant units to cope with ablations and weight decay pushing the weights of the network toward a smaller subspace.



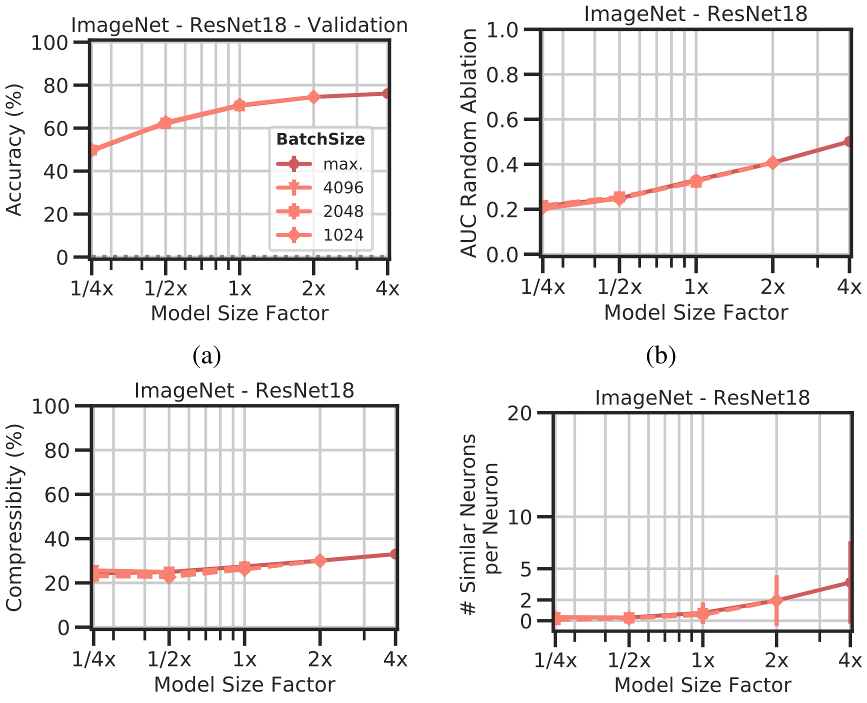
(a)

(b)

(c)

(d)

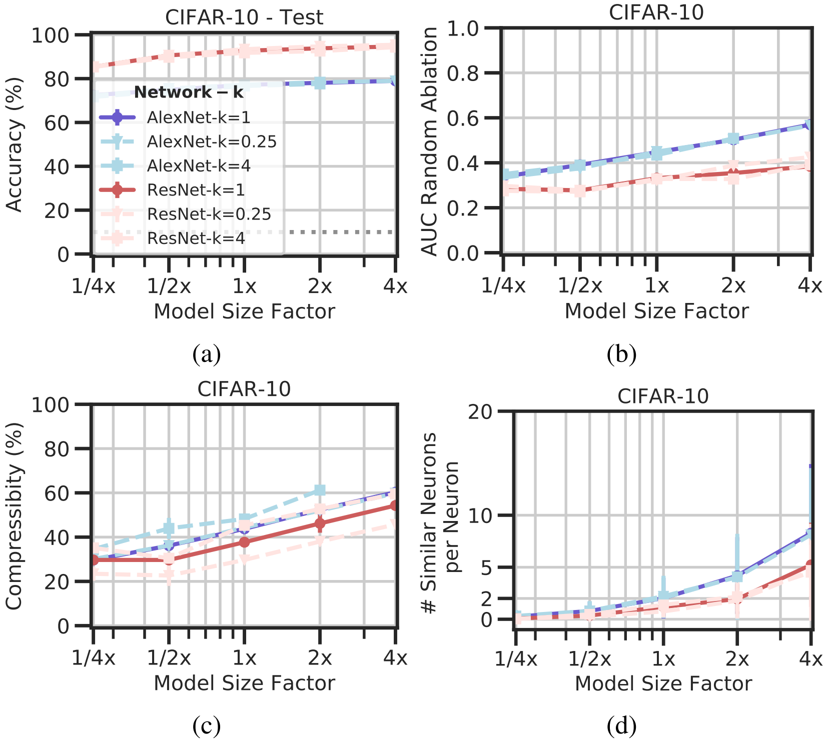
Figure C12: **Various optimizers result in different amounts but not different trends in robustness and redundancy in MLPs (Uncorrelated 10,000 dimensional synthetic data).** Trends in (a) accuracy, (b) robustness, (c) compressibility, and (d) similarity with momentum, stochastic gradient descent, and Adam optimizers across size factors for MLPs trained on 10,000 dimensional synthetic uncorrelated data. All three optimizers result in similar, flat curves in accuracy and robustness but increases for compressibility and similarity with size. Momentum results in a relatively high level of compressibility and similarity, SGD results in a low level of both, and Adam results in relatively low compressibility but high similarity. This may be a sign that these optimizers promote the development of silent/constant and redundant units to differing degrees.



(a) (b)

(c) (d)

Figure C13: **Robustness and redundancy do not depend on learning rate and batch size factor in ResNet18 (ImageNet).** Trends in (a) accuracy, (b) robustness, (c) compressibility, and (d) similarity across model sizes. We vary a constant factor k from 1=4 to 4 as a multiplier for the batch sizes and learning rates. "Max" refers to the maximum batch size that could be used for training a model given available hardware.



(a)

(b)

(c)

(d)

Figure C14: **The learning rate and batch size factor only has a slight effect on compressibility in AlexNets and ResNet56s (CIFAR-10).** Trends in (a) accuracy, (b) robustness, (c) compressibility, and (d) similarity across model sizes for AlexNets and ResNet56s. We vary a constant factor k from 1=4 to 4 as a multiplier for the batch sizes and learning rates. 4x AlexNets with $k = 4$ were not trained due to hardware restrictions. Compressibility tends to be slightly higher with higher k.