Deep Learning Generalization and the Convex Hull of Training Sets

Roozbeh Yousefzadeh
Yale University and VA Connecticut Healthcare System
roozbeh.yousefzadeh@yale.edu

Abstract

In this work, we study the generalization of deep learning functions in relation to the convex hull of their training sets. A trained image classifier basically partitions its domain via decision boundaries, and assigns a class to each of those partitions. The location of decision boundaries inside the convex hull of training set can be investigated in relation to the training samples. However, our analysis shows that in standard image classification datasets, most testing images are considerably outside that convex hull. Therefore, the performance of a trained model partially depends on how its decision boundaries are extended outside the convex hull of its training data. From this perspective, over-parameterization of deep learning models may be considered a necessity for shaping the extension of decision boundaries. At the same time, over-parameterization should be accompanied by a specific training regime, in order to yield a model that not only fits the training set, but also its decision boundaries extend desirably outside the convex hull. To illustrate this, we investigate the decision boundaries of a neural network, with various degrees of over-parameterization, inside and outside the convex hull of its training set. Moreover, we use a polynomial decision boundary to study the necessity of over-parameterization and the influence of training regime in shaping its extensions outside the convex hull of training set.

1 Introduction

A deep learning image classifier is a mathematical function that maps images to classes, i.e., a deep learning function [Strang, 2019]. These models/functions have shown to be exceptionally useful in real-world applications. However, generalization of these functions is considered a mystery by deep learning researchers [Arora et al., 2019]. These models have orders of magnitude more parameters than their training samples [Belkin et al., 2019, Neyshabur et al., 2019], and they can achieve perfect accuracy on their training sets, even when the training images are randomly labeled, or the contents of images are replaced with random noise [Zhang et al., 2017]. The training loss function of these models has infinite number of minimizers, where only a small subset of those minimizers generalize well [Neyshabur et al., 2017a]. If one succeeds in picking a good minimizer of training loss, the model can classify the testing images correctly, nevertheless, for any correctly classified image, there are infinite number of images that look the same, but models will classify them incorrectly (phenomenon known as adversarial vulnerability) [Papernot et al., 2016, Shafahi et al., 2019, Tsipras et al., 2019]. Here, we study some geometric properties of standard training and testing sets to provide new insights about what a model can learn from its training data, and how it can generalize.

Specifically, we study the convex hulls of image classification datasets (both in the pixel space and in the wavelet space), and show that most of testing images fall outside the convex hull of training sets, with various distances from the hull. We investigate the perturbation required to bring the testing

Deep Learning through Information Geometry Workshop
34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

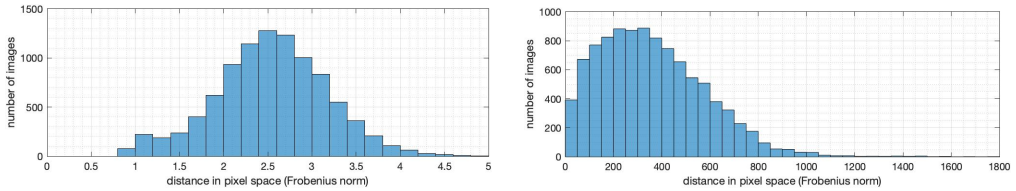
images to the surface of convex hull and observe that such perturbation significantly affects the contents of images. Therefore, the performance of a trained model partially depends on how well it can extrapolate. We investigate this extrapolation in relation to the over-parameterization of neural networks and the influence of training regimes in shaping the extensions of decision boundaries.

2 Geometry of testing data w.r.t the convex hull of training sets

First, we show that for standard datasets: MNIST [LeCun et al., 1998] and CIFAR-10 [Krizhevsky, 2009], most of their testing data are outside the convex hull of their training sets. We denote the convex hull of a training set by $\mathcal{H}^{tr}$.

To verify whether an image/data point is inside its corresponding $\mathcal{H}^{tr}$ or not, we can simply try to fit a hyper-plane separating the point and the training set. If we find such hyper-plane, the point is outside the convex hull, and vice versa. This is basically a linear regression problem and there are many efficient and fast methods to perform it, e.g., [Goldstein et al., 2015]. For the MNIST dataset, we see that about 95.1% of testing images are outside the $\mathcal{H}^{tr}$, in the pixel space. For CIFAR-10, that percentage is more than 99.9%. When we transform the images with wavelets (an operation analogous to convolutional neural nets), these percentages almost remain the same.

We can now investigate the testing data outside the $\mathcal{H}^{tr}$, to see how far they are located from it. One can measure the distance of a testing image to the $\mathcal{H}^{tr}$ using high-dimensional geometry algorithms. There are also methods that aim to identify a coreset to approximate convex hulls of large datasets [Blum et al., 2019]. Here, we approximate the distance to convex hull by first fitting a linear Support Vector Machine (SVM) between the testing point and the $\mathcal{H}^{tr}$. Since an SVM maximizes its margin from its supports, the total margin of the resulting SVM will closely approximate the distance between the point and $\mathcal{H}^{tr}$[1]. Figure 1 shows the histogram of distance to $\mathcal{H}^{tr}$ for the testing images of the above datasets.



(a) MNIST (pixel space) (b) CIFAR-10 (pixel space)

Figure 1: Variations of distance to $\mathcal{H}^{tr}$ for testing images that fall outside $\mathcal{H}^{tr}$.

To get a better sense of how far these distances are, consider the $\mathcal{H}^{tr}$ of CIFAR-10 dataset. Its diameter, the largest distance between any pair of vertices in $\mathcal{H}^{tr}$, is 13,621 (measured by Frobenius norm in pixel space). On the other hand, the distance of farthest testing image from the $\mathcal{H}^{tr}$ is about 1,800 (about 13% of the diameter of $\mathcal{H}^{tr}$). Moreover, the average distance between pairs of images in the training set of CIFAR-10 is 4,838, while the closest pair of images are only 701 apart.

Hence, the distance of testing data to $\mathcal{H}^{tr}$ is not negligible and we cannot dismiss it as a small noise. However, it is not very large either. Overall, we can say that in order to classify most of the testing images in the above datasets, a model has to extrapolate, to some degree, outside its $\mathcal{H}^{tr}$.

2.1 What it takes to bring an image inside the $\mathcal{H}^{tr}$

For each image that is outside the $\mathcal{H}^{tr}$, there is some minimum perturbation that would bring that image to the $\mathcal{H}^{tr}$. Figures 2 and 3 show the perturbation for some images in the testing set of CIFAR-10 and MNIST that can bring them to their corresponding $\mathcal{H}^{tr}$. We note that due to our approximation method, there is no guarantee that images in the middle column are exactly the minimum required perturbation, but we expect it to be sufficiently close to that minimum.

[1] In our experiments, we observe that in most cases, our computed SVMs are equidistant (or almost equidistant) from the testing points and the closest point of $\mathcal{H}^{tr}$. We note that this approximation of distance (i.e., using a linear SVM) does not overestimate the distance to $\mathcal{H}^{tr}$. In fact, the actual distances can be larger than the ones we report.

The perturbations required to bring testing images to the $\mathcal{H}^{tr}$ specifically relate to the objects of interest depicted in images and they appear to impact the images significantly. Therefore, the extrapolation required to classify those images can be considered significant, too.

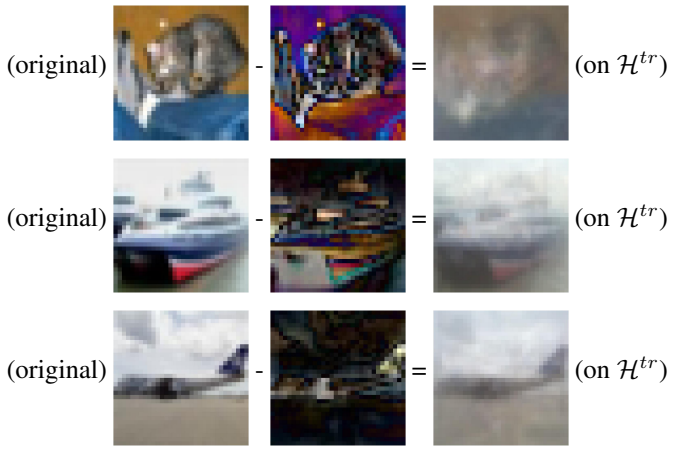


Figure 2: Perturbation (close to minimum) that can bring a testing image to $\mathcal{H}^{tr}$ of all classes. (left image) original testing image from CIFAR-10, (middle image) what should be removed from the original image, (right image) the resulting image on the $\mathcal{H}^{tr}$.

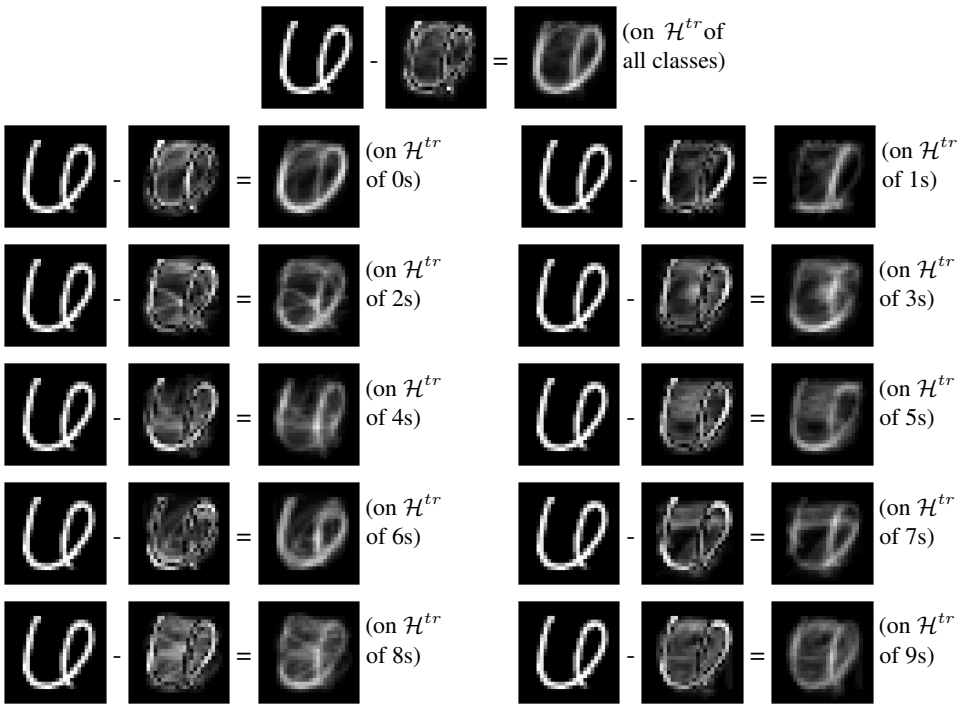


Figure 3: Perturbation that can bring a testing image of MNIST on the $\mathcal{H}^{tr}$.

2.2 Related work about geometry of data and deep learning

To the best of our knowledge, convex hulls of training sets are not commonly considered in deep learning studies, especially the ones focused on their generalization. Recently, Yousefzadeh and Huang [2020] reported that in the wavelet space, distance of testing images to the convex hull for each training class can predict the label for more than 98% of MNIST testing data. Previously, Haffner

[2002] considered the convex hull of MNIST data for Support Vector Machines. Similarly, Vincent and Bengio [2002] considered the convex hulls for K-Nearest Neighbor (KNN) algorithms. However, those methods do not generalize to deep learning functions.

Some researchers have studied other geometrical aspects of deep learning models, e.g., [Cohen et al., 2020, Fawzi et al., 2018, Cooper, 2018, Kanbak et al., 2018, Neyshabur et al., 2017b]. To our knowledge, those studies do not investigate the generalization of deep neural networks in relation to the convex hull of training sets. Most recently, Xu et al. [2020] studied the extrapolation behavior of ReLU perceptrons and concluded that such models cannot extrapolate most non-linear tasks. However, they do not connect their analysis to the fact that a considerable portion of testing samples of standard image datasets fall outside the convex hull of their training sets.

3 Learning outside the convex hull: A polynomial decision boundary

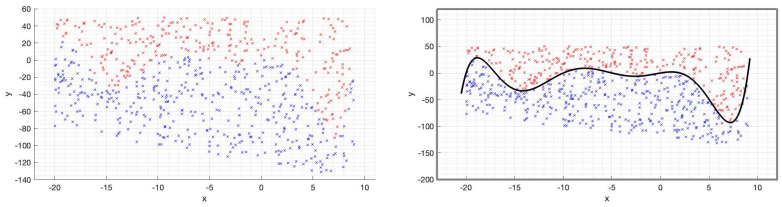
In the previous section, we showed that most of the testing data of MNIST and almost all of the testing data of CIFAR-10 are outside the convex hull of their corresponding training sets, while the distance to the $\mathcal{H}^{tr}$ has noticeable variations. Hence, a trained deep learning model somehow manages to define its decision boundaries accurately enough outside the boundaries of what it has observed during training. But how does a model achieve that, or more precisely, how do we manage to train a model such that its decision boundaries have the desirable form outside the $\mathcal{H}^{tr}$?

Since we are interested in the generalization of image classifiers, and the pixel space is a bounded space, we consider the domain to be bounded, while the $\mathcal{H}^{tr}$ occupies some portion of it. Testing data can be inside and outside the $\mathcal{H}^{tr}$, but always inside the bounded domain.

Let's now use a polynomial decision boundary as an example to gain some intuitive insights.[2] Figure 6a shows two point sets colored in blue and red, each set belonging to a class. These sets are non-linearly separable, because they have no overlap. If we use the polynomial

$$y = 10^{-5}(x+20)(x+17)(x+10)(x+5)(x)(x-2)(x-9), \tag{1}$$

as our decision boundary, we achieve perfect accuracy in separating these two sets, as shown in Figure 6b.



(a) (b)

Figure 4: **(a)** Training data with 2 classes, colored with blue and red. **(b)** Non-linear separation of 2 classes with a polynomial of degree 7.

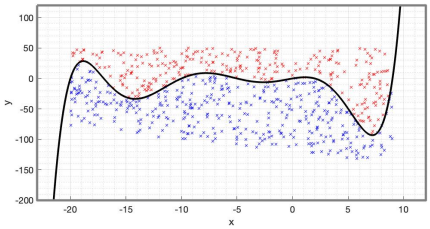


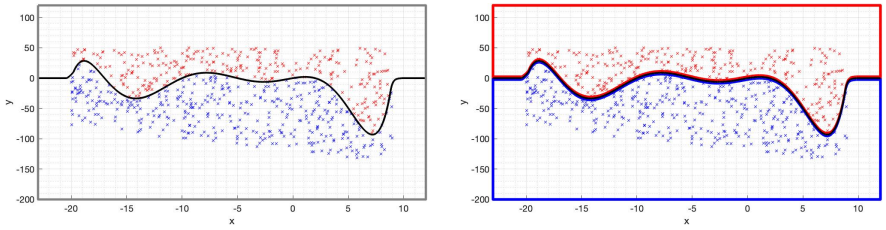
Figure 5: Shape of the polynomial decision boundary in our bounded domain, inside and outside the convex hull of its training data.

[2]This choice seem appropriate since many recent studies on generalization of deep learning consider the regression models that interpolate, e.g. [Belkin et al., 2018b,a, 2019, Liang et al., 2020, Verma et al., 2019, Kileel et al., 2019, Savarese et al., 2019], but those studies do not consider the convex hull of training sets.

Now that we have obtained this polynomial, i.e., decision boundary, we would be interested to know how it generalizes to unseen data. Let's assume that our bounded domain is defined by the limits shown in Figure 5 which also shows how our decision boundary generalizes outside the $\mathcal{H}^{tr}$. If our polynomial can correctly separate and label our testing data, we would say that our polynomial is generalizing well, and vice versa. But, what is reasonable to expect from the testing data? In what regions of the domain should we be hopeful that our polynomial can generalize? What if the domain is much larger than the $\mathcal{H}^{tr}$? Is the extension of our polynomial on both sides reasonable enough?

Clearly, the answer to the above questions can be different inside and outside the $\mathcal{H}^{tr}$. Inside the $\mathcal{H}^{tr}$, if the unseen data has a similar label distribution as the training set, we can be hopeful that our decision boundary will generalize well. However, outside the $\mathcal{H}^{tr}$ is uncharted territory and hence, there will be less hope/confidence about the generalization of our decision boundary, especially when we go far outside the $\mathcal{H}^{tr}$.

Now, let's assume that from some prior knowledge, we know that the decision boundary in Figure 6 is the unique decision boundary that perfectly classifies the testing data. In such case, the decision boundary defined by equation (1) and shown in Figure 5 will generalize poorly outside the $\mathcal{H}^{tr}$, despite the fact that it perfectly fits the training data.



(a) Possible extension of polynomial decision boundary in the over-parameterized regime.

(b) Resulting division of domain between the two classes, defined by red and blue bounds.

Figure 6: Consider the decision boundary depicted in **(a)** and assume that the distribution of testing data is such that the red and blue bounded regions in **(b)** are densely filled with red and blue data points, respectively. It follows that the decision boundary in Figure 5 generalizes poorly for testing points outside the $\mathcal{H}^{tr}$, despite the fact that it perfectly fits the training data.

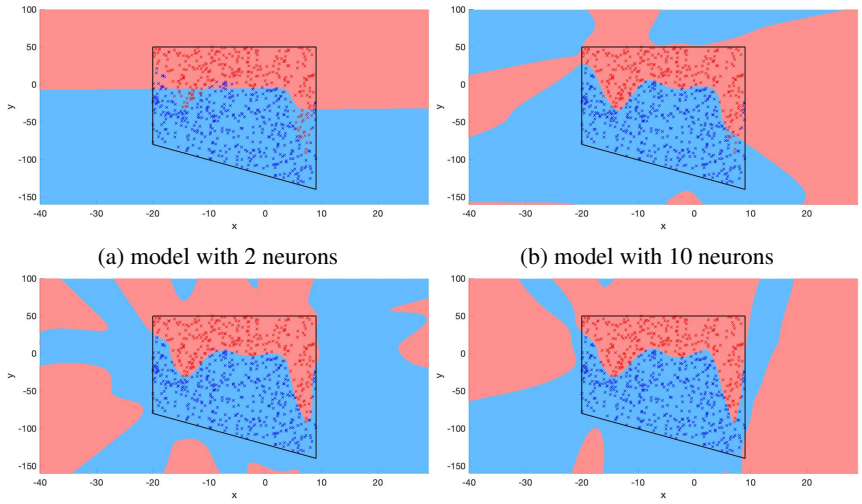
How can we incorporate that prior knowledge into the decision boundary defined by equation (1) and reshape it to the decision boundary in Figure 6, so that it can generalize well both inside and outside the $\mathcal{H}^{tr}$? How can we change the shape of our polynomial outside the $\mathcal{H}^{tr}$, while maintaining its current shape inside the $\mathcal{H}^{tr}$? Clearly, we should add to the degree of our polynomial, or in other words, we should **over-parameterize** it. The necessity of over-parameterization for achieving that goal for our polynomial decision boundary can be rigorously shown using the orthogonal system of Legendre polynomials [Ascher and Greif, 2011].

From this perspective, over-parameterization is necessary, but it is not sufficient for good generalization, because for an over-parameterized polynomial (i.e., decision boundary), there will be infinite number of solutions that can fit the training data, but each of them would have a different shape outside the $\mathcal{H}^{tr}$. In fact, an over-parameterized polynomial can have the same shape as the polynomial in Figure 5. But, how can we pick the decision boundary that fits the data and also generalizes well outside the $\mathcal{H}^{tr}$?

In the over-parameterized regime, the key to finding the desirable decision boundary is the optimization process, i.e., **the training regime**. In other words, different training regimes would lead us to decision boundaries that all perfectly fit the training set, but each has a different shape outside the $\mathcal{H}^{tr}$. This highlights that the over-parameterization and the training regime work in tandem to shape the extensions of our decision boundary.

4 Output of deep learning functions outside their $\mathcal{H}^{tr}$

In this section, we investigate a 2-layer neural network with ReLU activation. We train the model with various number of neurons on the data we investigate in previous section, depicted in Figure 6b. We then investigate the output of the trained models inside and outside of the $\mathcal{H}^{tr}$, as shown in Figure 7.



(a) model with 2 neurons

(b) model with 10 neurons

(c) model with 20 neurons

(d) model with 20 neurons

Figure 7: Variations of model output inside and outside the $\mathcal{H}^{tr}$, as we change the number of neurons in the model. The black trapezoid depicts the $\mathcal{H}^{tr}$. The colors red and blue correspond to our 2 classes, described earlier. The models (c) and (d) both have 20 neurons and achieve perfect accuracy on the training set, but they are trained using different hyper-parameters and as a result their decision boundaries outside the convex hull are completely different. Models with 2 and 10 neurons do not achieve perfect accuracy inside the convex hull.

We observe that when the model is under-parameterized (e.g., model with 2 neurons), the training regime does not have a significant effect on the resulting model. The model output also follows a relatively simple pattern inside and outside the $\mathcal{H}^{tr}$. For an under-parameterized model, the training loss function has limited number of minimizers, none of which lead to zero loss. Finding the same minimizer of training loss is equivalent to obtaining the same trained model, hence unlike the over-parameterized setting, the training regime is focused on finding the best shape for the decision boundary inside the convex hull.

When the model is over-parameterized, however, we have infinite number of parameter configurations that minimize the training loss to zero, which is equivalent to developing a decision boundary that perfectly separates our two sets of data points. So, forming the decision boundary inside the convex hull is easily achievable. What is different about those infinite number of models is the extension of their decision boundaries outside the convex hull of the data.

This seems to explain why we need over-parameterized models for deep learning and also explain why the generalization of deep learning models are so susceptible to different training regimes:

1. Generalization of deep learning models depends on how they extrapolate;
2. In order to desirably shape the extension of decision boundaries, we over-parameterize the models and then minimize their loss using specific training regimes.[3]

5 Conclusion and future work

We showed that most of testing data for some standard image classification models lie outside the convex hull of training sets, both in pixel space and in wavelet space. Therefore, the generalization of a deep network partially relies on its capability to extrapolate outside the boundaries of the data it has seen during training. Based on this observation, the significant number of studies that focus on interpolation regimes seem to be insufficient to explain the generalization of deep networks.

[3]Specific ways of minimizing the training loss imply the massive literature that aim to find the best training regime for achieving the best testing accuracy. Much of that literature has relied on the knowledge of testing/validation sets in order to develop those training methods, to desirably shape the extension of decision boundaries.

From this perspective, over-parameterization of models may be considered a necessity to desirably form the extension of decision boundaries outside the convex hull of data. This can be proven for polynomial regression models using the orthogonal system of Legendre polynomials. Moreover, we showed that the training regime can significantly affect the shape of decision boundaries outside the convex hulls, affecting the accuracy of a model in its extrapolation. We investigated a 2-layer ReLU network and a polynomial decision boundary to demonstrate these ideas.

In future work, we plan to more closely analyze the effect of over-parameterization and training regimes on the shape of decision boundaries outside the convex hulls, and investigate how that affects the generalization. We also plan to study how sensitive the classifications of standard trained models are w.r.t the minimum perturbations that would bring testing images inside the $\mathcal{H}^{tr}$. Developing efficient methods to compute that minimum perturbation could be useful.

Studying the convex hulls of internal representations of the data in a trained network is another direction that can be pursued. Such analysis can be performed, separately for each class in the dataset. It has been speculated that a given image classification dataset lies on a lower dimensional manifold and such manifold is what a deep learning model learns from the data. Study of convex hulls might provide insights about such manifold and also about the distribution of training and testing sets.

Finally, measuring the volume of the convex hulls of training and testing sets, their overlap, and also the volume of the domain that remains unoccupied may be insightful. The dimension of the domain (number of pixels) relative to the number of samples may have a significant effect on the portion of testing data that fall outside the $\mathcal{H}^{tr}$. This relates to the limit theorems for the convex hull of random points in higher dimensions [Hueter, 1999] and also to studies on separability and distribution of random points [Fink et al., 2016].

Acknowledgments and Disclosure of Funding

R.Y. thanks Yaim Cooper for helpful discussion. R.Y. was supported by a fellowship from the Department of Veteran Affairs. The views expressed in this manuscript are those of the author and do not necessarily reflect the position or policy of the Department of Veterans Affairs or the United States government.

References

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, pages 7413–7424, 2019.

Uri M Ascher and Chen Greif. *A first course on numerical methods*. SIAM, 2011.

Mikhail Belkin, Daniel J Hsu, and Partha Mitra. Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate. In *Advances in neural information processing systems*, pages 2300–2311, 2018a.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 541–549, 2018b.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Avrim Blum, Sariel Har-Peled, and Benjamin Raichel. Sparse approximation via generating point sets. *ACM Transactions on Algorithms (TALG)*, 15(3):1–16, 2019.

Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature communications*, 11(1):1–13, 2020.

Yaim Cooper. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200*, 2018.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2018.

Martin Fink, John Hershberger, Nirman Kumar, and Subhash Suri. Hyperplane separability and convexity of probabilistic point sets. In *32nd International Symposium on Computational Geometry (SoCG 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

Tom Goldstein, Min Li, and Xiaoming Yuan. Adaptive primal-dual splitting methods for statistical learning and image processing. In *Advances in Neural Information Processing Systems*, pages 2089–2097, 2015.

Patrick Haffner. Escaping the convex hull with extrapolated vector machines. In *Advances in Neural Information Processing Systems*, pages 753–760, 2002.

Irene Hueter. Limit theorems for the convex hull of random points in higher dimensions. *Transactions of the American Mathematical Society*, 351(11):4337–4363, 1999.

Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Geometric robustness of deep networks: Analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4441–4449, 2018.

Joe Kileel, Matthew Trager, and Joan Bruna. On the expressive power of deep polynomial neural networks. In *Advances in Neural Information Processing Systems*, pages 10310–10319, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

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

Tengyuan Liang, Alexander Rakhlin, et al. Just interpolate: Kernel "ridgeless" regression can generalize. *Annals of Statistics*, 48(3):1329–1347, 2020.

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

Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017b.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In *Conference on Learning Theory*, pages 2667–2690, 2019.

Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *International Conference on Learning Representations*, 2019.

Gilbert Strang. *Linear Algebra and Learning from Data*. Wellesley-Cambridge Press, 2019.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

Pascal Vincent and Yoshua Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In *Advances in neural information processing systems*, pages 985–992, 2002.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848*, 2020.

Roozbeh Yousefzadeh and Furong Huang. Using wavelets and spectral methods to study patterns in image-classification datasets. *arXiv preprint arXiv:2006.09879*, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.