Low Complexity Solutions For Interpolating Deep Neural Networks

Matt McManus
Department of Computer Science
MIT
mattmcm@mit.edu

Raunak Chowdhuri
Department of Computer Science
MIT
raunakc@mit.edu

Evan H Vogelbaum
Department of Computer Science
MIT
evanv@mit.edu

Abstract

A generally understood fact of classical learning theory is the tradeoff between bias and variance, often drawn in textbooks as a U-shaped curve with model complexity on the x-axis and generalization error on the y-axis [Luxburg and Schölkopf]. The theory posits that test error is related to the complexity of the hypothesis space from which we choose our model in a manner that has a single "sweet-spot" beyond which increasing model complexity increases test error. However this theory has been challenged in recent years by the remarkable success of deep neural network models. Such models are often highly overparameterized, with parameter counts sometimes ranging in the millions to billions [A. Canziani, 2016]. Since free parameter count is a common measure of model complexity, one would expect these models to dramatically overfit to the training data and as a result the model selected from a procedure like ERM will not generalize well to unseen data. Surprisingly, however, the opposite behavior is often seen in what has been characterized as a "double descent curve." In this empirically observed regime, increasing model complexity beyond the point of interpolation (the point at which there exists models in the hypothesis space capable of achieving near 0 error on the training data) can lead to the selection of models that are able to generalize **better** than those selected from hypothesis spaces with complexity below the interpolation threshold. This remarkable result has been documented across several empirical studies and is summarized in detail by Belkin et al. [2019]. One important result from Belkin et al. [2019]'s analysis is that minimum norm (low complexity) interpolating solutions tend to be the best performing ones. In this project, we aim to further extend Belkin et al. [2019]'s work to the regime of deep neural networks. We develop a novel algorithm for learning low-complexity deep neural networks and evaluate it against baselines. Our results show that for certain classes of models our algorithm can produce comparatively better deep neural networks, and that across a variety of algorithms lower complexity deep neural network lead to lower complexity intermediate features. These results provide further evidence that small norm interpolating models have very desirable properties which merit further exploration.

1 Introduction

1.1 Bias Variance Tradeoff and the Classical Regime

In the classical machine learning regime we seek to increase the complexity of our hypothesis space to find a model that will have both the capacity required to learn our target function (bias) and have predictions that are robust to small changes in the training set and generalize well (variance). This is commonly accomplished by looking for a "sweet spot" when examining models from hypothesis classes of increasing complexity (such as those with more free parameters)[Belkin et al., 2019].

However recent results such as those characterized in Belkin et al. [2019] have shown that overparameterizing our model far beyond the capacity required to fully fit a training dataset can lead to even better generalization than the "balanced" model found in the classical regime. This confounding of the classical bias variance tradeoff has been the subject of considerable study [Zhang et al., 2016, Belkin et al., 2018, Yang et al., 2020, Geman et al., 1992]. Previous work has sought to understand the properties of so-called "interpolating" models–models which are able to completely fit the training dataset and have far more parameters than are required to do so. Belkin et al. [2019] found that interpolating models had better generalization error than classical models across a wide variety of tasks and model types. They detailed both empirical and theoretical results implying that the best interpolating models are minimal norm–low complexity–ones. Similar results have been published by Rangamani et al. [2020], who showed that minimal norm interpolating kernel machines have very desirable properties such as maximum stability. In our work, we aim to extend these results to the setting of deep neural networks.

1.2 Previous Work on Low Complexity Deep Neural Networks

A variety of previous studies have explored the properties low-complexity deep neural networks as well as overparameteterized neural networks, a pre-requisite for interpolation. Bansal et al. [2018] evaluates a method for training deep neural networks by regulating their complexity using an L2 penalty and lagrange multipliers, similiar to the training of SVMs. E et al. [2021] analyzes the generalization properties of minimum norm overparameterized networks and is able to derive bounds on their generalization error in the case of noiseless labels. Zhang et al. [2016] showed that overparameterized deep neural networks are able to express any function on a finite input sample. Huh et al. [2021] developed theoretical and empirical analysis showing that over-parameterized deep neural networks have an implicit bias towards low-rank feature embeddings. Most relevant to our work, Belkin et al. [2019] demonstrates the double descent phenomenon in a wide variety of contexts and posits that low-complexity solutions are the best ones in the modern "interpolating" regime. In our work we seek to extend that of Belkin et al. [2019] by using the complexity measure of effective rank [Roy and Vetterli, 2007], inspired by its use in Huh et al. [2021]. From here on out, when we discuss complexity, we refer to the measure of the average effective rank of the weight matrices in a deep neural network.

1.3 Projection Algorithm

Algorithm 1: Projection Algorithm

```
Input: Interpolating solution $f_\theta$, with parameters $\theta$
Output: $\theta_{\text{min}}$, an approximation of the minimum effective rank interpolating solution
Initialize $\theta_{\text{min}} = \theta$
while Not should_terminate($\theta, \theta_{min}$) do
    $\theta_{\text{min}} = \theta$
    while $\mathcal{L}(f_\theta, \mathcal{D}_{train}) = 0$ do
        $U\Sigma V = \text{SVD}(\theta)$                        // SVD decomposition
        $\theta = U\,\text{zero\_last\_column}(\Sigma)V$       // reduce rank by one
    while $\mathcal{L}(f_\theta, \mathcal{D}_{train}) \neq 0$ do
        $\theta = \theta - \alpha \cdot \frac{\partial \mathcal{L}}{\partial \theta}$       // Take gradient step
```

In order to facilitate the training of low-effective rank interpolating neural networks, we introduce a new algorithm inspired by a constrained optimization method in adversarial robustness called projected gradient descent (PGD) [Madry et al., 2017]. The idea is to use a traditional optimization method until the constraints are no longer satisfied. Once this occurs, we can project back into valid constraint space and continue performing optimization.

In our algorithm, we impose the constraint that the model is an interpolating one, and we seek to minimize the average effective rank of all of the linear weight matrices in the network. The algorithm is detailed in Algorithm 1. First, we train a network to the point of interpolation using standard ERM training methods (e.g. gradient descent). In practice, we have found training slightly beyond this point produces better results (we treat the number of training steps beyond the point of interpolation as a hyperparameter). Once we have finished training, we then reduce the effective rank of each layer of the network by computing the SVD of its weight matrix and zeroing out the smallest singular value. We continue this rank-reduction process until our model no longer meets a threshold for interpolation (in practice we find that 85% accuracy works well as a stopping point). We then repeat the process, alternating training to interpolation and reducing effective rank until termination.

2 Experiments

2.1 Training Method Comparison

We aim to study the relationship between the utilized method of regularizing effective rank and its effect on training accuracy and effective rank of the neural networks. We evaluate this relationship on four classes of interpolating (weakened to train accuracy being $> 95\%$) models:

- Vanilla MLP: A standard MLP network with 2000 parameters. We construct 10 different MLPs with the number of hidden layers ranging from 1 to 10. Hidden layer widths are adjusted to keep the parameter count as close to 2000 as possible so that our models have a fixed level of capacity.
- All Layer Regularization: The same architecture as the Vanilla MLP but with a penalty term added to the loss based on the effective rank of the network. To make regularization strength comparable across architectures, we average the effective rank of each layer's weight matrix divided by the maximum rank of the weight matrix across all layers in the network.
- Last Layer Regularization: Since the last layer collects the features of a neural network and combines them to produce outputs, we hypothesize that only regularizing the last layer may still lead to a regularizing effect, as a simple final layer must work with simpler features to produce its outputs. This class of models is identical to the All Layer Regularization class but only applies regularization to the final layer.
- Projection All Layers: This class of models uses architectures identical to those of the Vanilla MLP but is trained using the projection algorithm (Algorithm 1) applied to all weight matrices
- Projection Last Layer: Using similar reasoning as the Last Layer Regularization models, we also try only applying our projection algorithm to the last layer of our architecture.

Models are trained across 3 random seeds for 5000 steps and the reported results are averaged. For all experiments, we use the MNIST-1D dataset [Greydanus, 2020], a simpler version of MNIST that maintains the real world complexity of the dataset with a significantly smaller feature dimension, allowing for the evaluation of smaller models.

2.2 Feature Complexity Analysis

In [Huh et al., 2021], it was found that deep neural networks have an inductive bias towards low rank feature embeddings. We hypothesize that low-complexity interpolating networks may have the same bias as having simpler weights should portend having simpler features since the model cannot easily learn high variance weights which overfit to small details of the data. To test this hypothesis, we evaluate the feature complexity for the same four classes of interpolating models.

To measure feature complexity we leverage the approach used in [Huh et al., 2021]: the feature complexity of a model is the effective rank of the kernel matrix of the features the model produces on

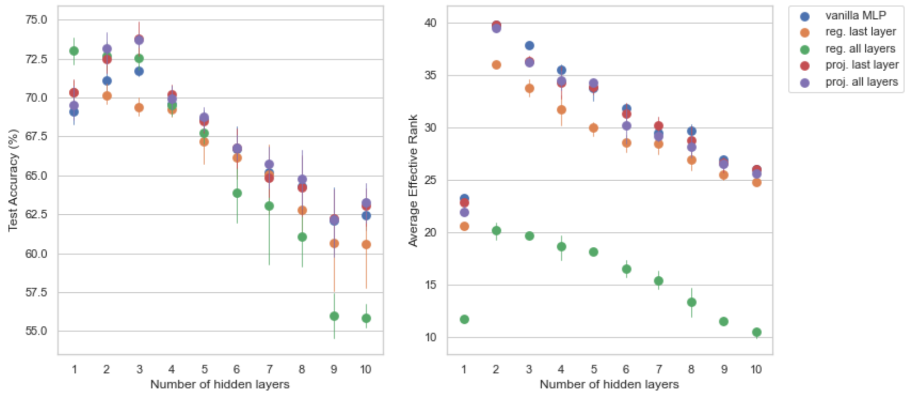

Figure 1: **(Left)** A comparison of generalization accuracy for five different classes of models with varying depth. **(Right)** A comparison of Effective Rank for five different classes of models with varying depth.

input data. We use the features produced right before the final layer (after the ReLu activation). For computational reasons, we evaluate feature complexity on a random subsample of 100 datapoints in our training set. We find that relationships shown using larger sample sizes are maintained with this sample size.

3 Results and Analysis

3.1 Effective Rank and Test Accuracy

In Figure 1, we show a comparison of test accuracy and effective rank for all five classes of models we consider. In the left plot, we see that for a small number of hidden layers (2 to 3), our projection based methods are able to on average outperform all other methods including the unregularized vanilla MLP. As depth increases, however, the generalization error decreases and our average performance becomes close to identical to that of the unregularized vanilla MLP. Notably, however, we still generalize better than the baseline explicitly regularized models. This shows the potential of our projection algorithm for training high performing low effective rank models.

On the right, we see a comparison of the average effective rank of each class of models as we increase depth. While the effective rank of our algorithm is consistently lower than that of the vanilla MLP on average, explicit regularization seems far more effective and reducing effective rank.

3.2 Feature Complexity

We show results for our feature complexity experiment in Figure 2. On the left, we show how the feature complexity changes as we increase the number of layers in our model (keeping the number of parameters fixed) for all five classes of models we tested. As expected based on the results of [Huh et al., 2021], we see that increasing the depth of the network dramatically reduces the complexity of features our network learns. Surprisingly, our projection method seems to learn the most complex features across all model architectures we consider outside of the deepest (9 and 10 layer networks). We hypothesize that this may be a result of the projection method, which seeks to reduce rank by removing the smallest singular value components of a matrix first. Since we are removing the smallest components first, these components will have the least impact on the overall complexity of the features output by the model. Therefore the projection method is able to maintain the structural complexity of features that a deep neural network needs to learn relationships in the data while filtering out the spurious complexity which comes as a result of noise. A complicating and suprising observation we made during training is that effective rank of features tended to **increase** as the effective rank of our network decreased when we trained using our projection method. This result was not unique to our projection method, but was particularly pronounced in models that were trained with it, especially lower depth models. This could serve as an interesting point of exploration in future work.

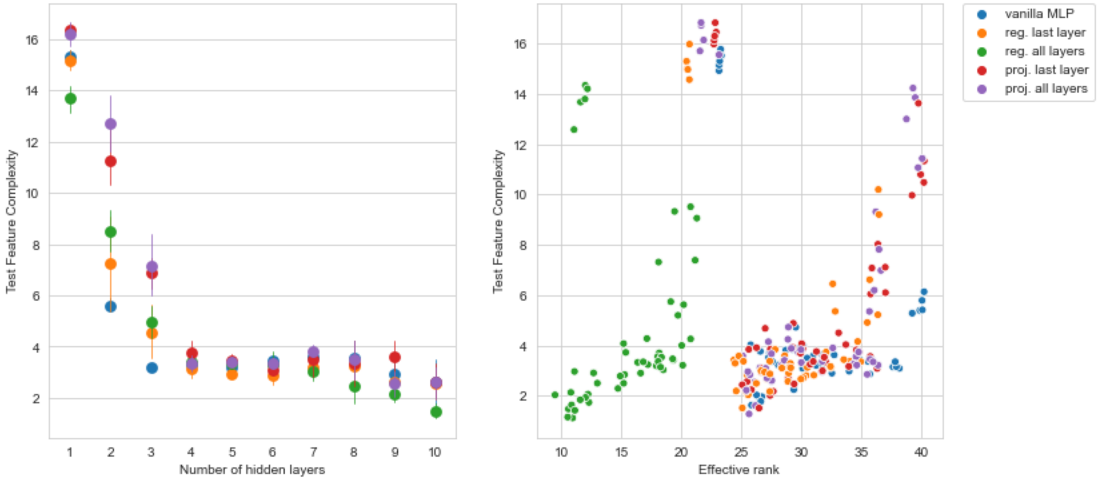

Figure 2: **(Left)** A comparison of feature complexity for five different classes of models with varying depth. Experiments are run across 3 seeds and error bars show 95% confidence intervals. **(Right)** A scatterplot of effective rank and feature complexity for all five classes of models in our Feature Complexity experiment across seed and number of layers.

On the right of Figure 2, we show a scatterplot of effective rank versus feature complexity for all models and seeds that we trained. The trend is clear across almost all sets of models that we train: lower effective rank leads to much lower feature complexity. The only exception to this rule are the 1-layer and some 2-layer Neural Networks, which we hypothesize do not have enough depth to simplify the high dimensional input data sufficiently before the final layer of the network.

4 Conclusion and Future Work

As detailed in [Belkin et al., 2019], interpolating machine learning models seem to have many surprising and desirable properties, and this paper presents one more step towards understanding these properties. We have developed a new algorithm to find minimum effective rank interpolating deep neural networks and shown that for certain model architectures it is able to outperform strong baselines. We further analyze the relationship between effective rank of a network's weights and the complexity of its features and find that small effective rank weights produce less complex features. These results extend those of Huh et al. [2021]. We think that future work extending our analysis to larger models and dataset will provide a better understanding of the properties of low-complexity interpolating deep networks.

5 Acknowledgements

We would first like to thank the 6.867 teaching staff for all of their work throughout the semester. We would like to thank Professor Agrawal and Yilun Du in particular for very helpful feedback on our project at every step of the process.

References

Ulrike von Luxburg and Bernhard Schölkopf. Statistical learning theory: Models, concepts, and results. In Dov M. Gabbay, Stephan Hartmann, and John Woods, editors, *Handbook of the History of Logic*, volume 10 of *Inductive Logic*, pages 651–706. North-Holland. doi: 10.1016/B978-0-444-52936-7.50016-1. URL https://www.sciencedirect.com/science/article/pii/B9780444529367500161.

E. Culurciello A. Canziani, A. Paszke. An analysis of deep neural network models for practical applications. 2016.

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.

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

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 541–549. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/belkin18a.html.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks, 2020.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, 01 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.1.1. URL https://doi.org/10.1162/neco.1992.4.1.1.

Akshay Rangamani, Lorenzo Rosasco, and Tomaso Poggio. For interpolating kernel machines, minimizing the norm of the erm solution minimizes stability, 2020.

Yamini Bansal, Madhu Advani, David D Cox, and Andrew M Saxe. Minnorm training: an algorithm for training over-parameterized deep neural networks, 2018.

Weinan E, Chao Ma, and Lei Wu. The generalization error of the minimum-norm solutions for over-parameterized neural networks, 2021.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks, 2021.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. 01 2007.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Sam Greydanus. Scaling down deep learning, 2020.