Provable Benefits of Overparameterization in Model Compression: From Double Descent to Pruning Neural Networks

Xiangyu Chang[1], **Yingcong Li**[1], **Samet Oymak**[1] and **Christos Thrampoulidis**[2*]
[1] University of California, Riverside
[2] University of British Columbia
{cxian008, yli692, soymak}@ucr.edu, cthrampo@ece.ubc.ca

Abstract

Deep networks are typically trained with many more parameters than the size of the training dataset. Recent empirical evidence indicates that the practice of overparameterization not only benefits training large models, but also assists – perhaps counterintuitively – building lightweight models. Specifically, it suggests that overparameterization benefits model pruning / sparsification. This paper sheds light on these empirical findings by theoretically characterizing the high-dimensional asymptotics of model pruning in the overparameterized regime. The theory presented addresses the following core question: "should one train a small model from the beginning, or first train a large model and then prune?". We analytically identify regimes in which, even if the location of the most informative features is known, we are better off fitting a large model and then pruning rather than simply training with the known informative features. This leads to a new double descent in the training of sparse models: growing the original model, while preserving the target sparsity, improves the test accuracy as one moves beyond the overparameterization threshold. Our analysis further reveals the benefit of retraining by relating it to feature correlations. We find that the above phenomena are already present in linear and random-features models. Our technical approach advances the toolset of high-dimensional analysis and precisely characterizes the asymptotic distribution of over-parameterized least-squares. The intuition gained by analytically studying simpler models is numerically verified on neural networks.

1 Introduction

Large model size and overparameterization in deep learning are known to improve generalization performance (Neyshabur et al. 2017), and, state-of-the-art deep neural networks (DNNs) can be outrageously large. However, such large models are not suitable for certain important application domains, such as mobile computing (Tan et al. 2019; Sandler et al. 2018). Pruning algorithms aim to address the challenge of building lightweight DNNs for such domains. While there are several pruning methods, their common goal is to compress large DNN models by removing weak connections/weights with minimal decline in accuracy. Here, a key empirical phenomenon is that *it is often better to train and prune a large model rather than training a small model from scratch*. Unfortunately, the mechanisms behind this phenomenon are poorly understood especially for practical gradient-based algorithms. This paper sheds light on this by answering: *What are the optimization and generalization dynamics of pruning overparameterized models? Does gradient descent naturally select the good weights?*

Contributions: We analytically study the performance of popular pruning strategies. First, we analyze linear models, and then, generalize the results to nonlinear feature maps. Through extensive simulations, we show that our analytical findings predict similar behaviors in more complex settings.
(a) Distributional characterization (DC): The key innovation facilitating our results is a theoretical characterization of the distribution of the solution of overparameterized least-squares. This DC enables us to accurately answer *"what happens to the accuracy if X% of the weights are pruned?"*.
(b) Benefits of overparameterization: Using DC, we obtain rigorous precise characterizations of the pruning performance in linear problems. Furthermore, we use, so called "linear gaussian equivalences", to obtain sharp analytic predictions for nonlinear maps, which we verify via extensive numerical simulations. By training models of growing size and compressing them to fixed sparsity, we identify a novel double descent behavior, where the risk of the pruned model is consistently minimized in the overparameterized regime. Using our theory, we uncover rather surprising scenarios where pruning an overparameterized model is provably better than training a small model with the exact information of optimal nonzero locations.
(c) Benefits of retraining: An important aspect of pruning is retraining the model using the favorable nonzero locations identified during the initial training. We show that retraining can actually hurt the performance when features are uncorrelated. However, it becomes critical as correlations increase. Importantly, we devise the DC of the *train→prune→retrain process* (see Figs. 2 and 4 and the discussion around Def. 5 for details), and, we demonstrate that it correctly captures the pruning performance of random features that are known to be good proxies for understanding DNN behavior (Jacot, Gabriel, and Hongler 2018).

We anticipate that our techniques towards establishing the DC of the overparameterized problems might be useful, beyond the context of pruning, in other statistical inference

[*] The author names are in alphabetical order.

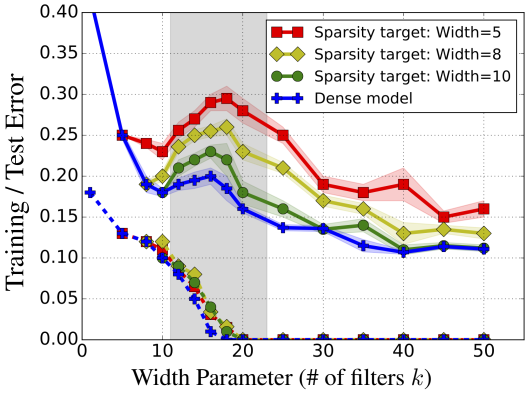


Figure 1: We train sparse ResNet-20 models on the CIFAR-10 dataset with varying width (i.e. # of filters) and sparsity targets. The solid and dashed lines are test and training errors respectively. The shaded region highlights the transition to zero training error.

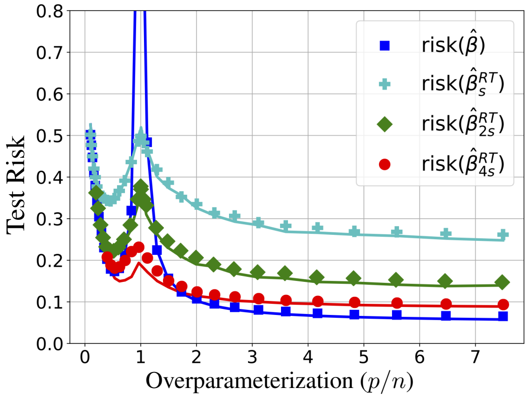


Figure 2: Random feature regression (RFR) and pruning with ReLU feature map ($\phi(\boldsymbol{a}) = \text{ReLU}(\boldsymbol{Ra})$) and varying sparsity targets. Solid lines follow from our distributional characterization and the markers are obtained by solving the actual RFR.

tasks that require careful distributional studies.

1.1 Prior Art

This work relates to the literature on model compression and overparameterization in deep learning.

Neural network pruning: Large model sizes in deep learning have led to a substantial interest in model pruning/quantization (Han, Mao, and Dally 2015; Hassibi and Stork 1993; LeCun, Denker, and Solla 1990). DNN pruning has a diverse literature with various architectural, algorithmic, and hardware considerations (Sze et al. 2017; Han et al. 2015). The pruning algorithms can be applied before, during, or after training a dense model (Lee, Ajanthan, and Torr 2018; Wang, Zhang, and Grosse 2020; Jin et al. 2016; Oymak 2018) and in this work we focus on after training. Related to over-parameterizarion, (Frankle and Carbin 2019) shows that a large DNN contains a small subset of favorable weights (for pruning), which can achieve similar performance to the original network when trained with the same initialization. (Zhou et al. 2019; Malach et al. 2020; Pensia et al. 2020) demonstrate that there are subsets with good test performance even without any training and provide theoretical guarantees. However, these works do not answer why practical gradient-based algorithms lead to good pruning outcomes. Closer to us, (Li et al. 2020) derives formulas for predicting the pruning performance of over-parameterized least-squares without proofs. In contrast, we provide provable guarantees, and, also obtain DC for more complex problems with general design matrices and nonlinearities.

Benefits of overparameterization: Studies on the optimization and generalization properties of DNNs demonstrate that overparameterization acts as a catalyst for learning. (Arora, Cohen, and Hazan 2018; Neyshabur, Tomioka, and Srebro 2014; Gunasekar et al. 2017; Ji and Telgarsky 2018) argue that gradient-based algorithms are implicitly biased towards certain favorable solutions (even without explicit regularization) to explain benign overfitting (Bartlett et al. 2020; Oymak and Soltanolkotabi 2020; Du et al. 2018; Chizat, Oyallon, and Bach 2019; Belkin, Ma, and Mandal 2018; Belkin, Rakhlin, and Tsybakov 2019; Tsigler and Bartlett 2020; Liang and Rakhlin 2018; Mei and Montanari 2019; Ju, Lin, and Liu 2020). More recently, these studies have led to interesting connections between kernels and DNNs and a flurry of theoretical developments. Closest to us, (Nakkiran et al. 2019; Belkin, Hsu, and Xu 2019; Belkin et al. 2019) uncover a double-descent phenomenon: the test risk has two minima as a function of model size. One minimum occurs in the classical underparameterized regime whereas the other minimum occurs when the model is overparameterized and the latter risk can in fact be better than former. Closer to our theory, (Dereziński, Liang, and Mahoney 2019; Hastie et al. 2019; Montanari et al. 2019; Deng, Kammoun, and Thrampoulidis 2019; Kini and Thrampoulidis 2020; Liang and Sur 2020; Salehi, Abbasi, and Hassibi 2020; Ju, Lin, and Liu 2020) characterize the asymptotic performance of overparameterized learning problems. However these works are limited to characterizing the test error of regular (dense) training. In contrast, we use distributional characterization (DC) to capture the performance of more challenging pruning strategies and we uncover novel double descent phenomena (see Fig. 1).

2 Problem Setup

Let us fix the notation. Let $[p] = \{1, 2, \ldots, p\}$. Given $\boldsymbol{\beta} \in \mathbb{R}^p$, let $\mathbb{T}_s(\boldsymbol{\beta})$ be the pruning operator that sets the smallest $p - s$ entries in absolute value of $\boldsymbol{\beta}$ to zero. Let $\mathcal{I}(\boldsymbol{\beta}) \subset [p]$ return the index of the nonzero entries of $\boldsymbol{\beta}$. $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. $\boldsymbol{X}^\dagger$ denotes the pseudoinverse of matrix $\boldsymbol{X}$.

Data: Let $(\boldsymbol{a}_i, y_i)_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$ with i.i.d. input-label pairs. Let $\phi(\cdot) : \mathbb{R}^d \to \mathbb{R}^p$ be a (nonlinear) feature map. We generate $\boldsymbol{x}_i = \phi(\boldsymbol{a}_i)$ and work with the dataset $\mathcal{S} = (\boldsymbol{x}_i, y_i)_{i=1}^n$ coming i.i.d. from some distribution $\mathcal{D}$. As an example, of special interest to the rest of the paper, consider random feature regression, where $\boldsymbol{x}_i = \psi(\boldsymbol{Ra}_i)$ for a nonlinear activation function ψ that acts entry-wise and a random matrix $\boldsymbol{R} \in \mathbb{R}^{p\times d}$ with i.i.d. $\mathcal{N}(0,1)$ entries; see Fig. 2. In matrix notation, we let $\boldsymbol{y} = [y_1 \dots y_n]^T \in \mathbb{R}^n$ and $\boldsymbol{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_n]^T \in \mathbb{R}^{n\times p}$ denote the vector of labels and the feature matrix, respectively. Throughout, we focus on regression tasks, in which the training and the test risks of a model $\boldsymbol{\beta}$ are defined as

$$\text{Population risk:} \quad \mathcal{L}(\boldsymbol{\beta}) = \mathbb{E}_{\mathcal{D}}[(y - \boldsymbol{x}^T\boldsymbol{\beta})^2]. \tag{1}$$

$$\text{Empirical risk:} \quad \hat{\mathcal{L}}(\boldsymbol{\beta}) = \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_{\ell_2}^2. \tag{2}$$

During training, we will solve the empirical risk minimization (ERM) problem over a set of selected features $\Delta \subset [p]$, from which we obtain the least-squares solution

$$\hat{\boldsymbol{\beta}}(\Delta) = \arg \min_{\boldsymbol{\beta}\,:\,\mathcal{I}(\boldsymbol{\beta})=\Delta} \hat{\mathcal{L}}(\boldsymbol{\beta}). \tag{3}$$

For example, regular ERM corresponds to $\Delta = [p]$, and we simply use $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}([p])$ to denote its solution above. Let $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$ be the covariance matrix and $\boldsymbol{b} = \mathbb{E}[y\boldsymbol{x}]$ be the cross-covariance. The parameter minimizing the test error is given by $\boldsymbol{\beta}^\star = \boldsymbol{\Sigma}^{-1}\boldsymbol{b}$. We are interested in training a model over the training set $\mathcal{S}$ that not only achieves small test error, but also, it is sparse. We do this as follows. First, we run stochastic gradient descent (SGD) to minimize the empirical risk (starting from zero initialization). It is common knowledge that SGD on least-squares converges to the minimum ℓ_2 norm solution given by $\hat{\boldsymbol{\beta}} = \boldsymbol{X}^\dagger\boldsymbol{y}$. Next, we describe our pruning strategies to compress the model.

Pruning strategies: Given dataset $\mathcal{S}$ and target sparsity level s, a pruning function P takes a model $\boldsymbol{\beta}$ as input and outputs an s-sparse model $\boldsymbol{\beta}_s^P$. Two popular pruning functions are magnitude-based (MP) and Hessian-based (HP) (a.k.a. optimal brain damage) pruning (LeCun, Denker, and Solla 1990). The latter uses a diagonal approximation of the covariance via $\hat{\boldsymbol{\Sigma}} = \text{diag}(\boldsymbol{X}^T\boldsymbol{X})/n$ to capture *saliency* (see (4)). Formally, we have the following definitions:

- *Magnitude-based pruning:* $\boldsymbol{\beta}_s^M = \mathbb{T}_s(\boldsymbol{\beta})$.
- *Hessian-based pruning:* $\boldsymbol{\beta}_s^H = \hat{\boldsymbol{\Sigma}}^{-1/2}\mathbb{T}_s(\hat{\boldsymbol{\Sigma}}^{1/2}\boldsymbol{\beta})$.
- *Oracle pruning:* Let $\Delta^\star \subset [p]$ be the optimal s indices so that $\hat{\boldsymbol{\beta}}(\Delta^\star)$ achieves the minimum population risk (in expectation over $\mathcal{S}$) among all $\hat{\boldsymbol{\beta}}(\Delta)$ and any subset Δ in (3). When $\boldsymbol{\Sigma}$ is diagonal and $s < n$, using rather classical results, it can be shown that (see Lemma 7 in the extended version (Chang et al. 2020)) these *oracle indices* are the ones with the top-s saliency score given by

$$\text{Saliency score} = \boldsymbol{\Sigma}_{i,i}\boldsymbol{\beta}_i^{\star 2}. \tag{4}$$

Oracle pruning employs these latent saliency scores and returns $\boldsymbol{\beta}_s^O$ by pruning the weights of $\boldsymbol{\beta}$ outside of $\Delta^\star$.

We remark that our distributional characterization might allow us to study more complex pruning strategies, such as optimal brain surgeon (Hassibi, Stork, and Wolff 1994). However, we restrict our attention to the three aforementioned core strategies to keep the discussion focused.

Pruning algorithm: To shed light on contemporary pruning practices, we will study the following three-stage *train→prune→retrain* algorithms.

1. Find the empirical risk minimizer $\hat{\boldsymbol{\beta}} = \boldsymbol{X}^\dagger\boldsymbol{y}$.
2. Prune $\hat{\boldsymbol{\beta}}$ with strategy P to obtain $\hat{\boldsymbol{\beta}}_s^P$.
3. *Retraining:* Obtain $\hat{\boldsymbol{\beta}}_s^{RT} = \hat{\boldsymbol{\beta}}(\mathcal{I}(\hat{\boldsymbol{\beta}}_s^P))$.

The last step obtains a new s-sparse model by solving ERM in (3) with the features $\Delta = \mathcal{I}(\hat{\boldsymbol{\beta}}_s^P)$ identified by pruning. Figures 1 and 2 illustrate the performance of this procedure for ResNet-20 on the CIFAR-10 dataset and for a random feature regression on a synthetic problem, respectively . Our analytic formulas for RF, as seen in Fig. 2, very closely match the empirical observations (see Sec. 3 for further explanations). Interestingly, the arguably simpler RF model already captures key behaviors (double-descent, better performance in the overparameterized regime, performance of sparse model comparable to large model) in ResNet.

Sections 3 and 4 present numerical experiments on pruning that verify our analytical predictions, as well as, our insights on the fundamental principles behind the roles of overparameterization and retraining. Sec 5 establishes our theory on the DC of $\hat{\boldsymbol{\beta}}$ and provable guarantees on pruning. All proofs are deferred to the extended version (Chang et al. 2020).

3 Motivating Examples

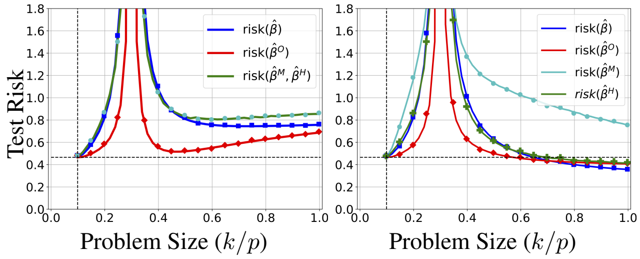
3.1 Linear Gaussian Problems

We begin our study with linear Gaussian problems (LGP), which we formally define as follows.

Definition 1 (Linear Gaussian Problem (LGP)) *Given latent vector $\boldsymbol{\beta}^\star \in \mathbb{R}^d$, covariance $\boldsymbol{\Sigma}$ and noise level σ, assume that each example in $\mathcal{S}$ is generated independently as $y_i = \boldsymbol{x}_i^T\boldsymbol{\beta}^\star + \sigma z_i$ where $z_i \sim \mathcal{N}(0,1)$ and $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. Additionally, the map $\phi(\cdot)$ is identity and $p = d$.*

Albeit simple, LGPs are of fundamental importance for the following reasons: (1) We show in Sec. 5 that our theoretical framework rigorously characterizes pruning strategies for LGPs; (2) Through a "linear Gaussian equivalence", we will use our results for linear models to obtain analytic predictions for nonlinear random features; (3) Our theoretical predictions and numerical experiments discussed next demonstrate that LGPs already capture key phenomena observed in more complex models (e.g., Fig. 1).

In Fig. 3, we consider LGPs with diagonal covariance $\boldsymbol{\Sigma}$. We set the sparsity level $s/p = 0.1$ and the relative dataset size $n/p = 0.3$. To parameterize the covariance and $\boldsymbol{\beta}^\star$, we use a *spiked* vector $\boldsymbol{\lambda}$, the first s entries of which are set equal to $C = 25 \gg 1$ and the remaining entries equal to 1. $\boldsymbol{\lambda}$ corresponds to the latent saliency score (cf. (4)) of the indices. To understand the role of overparameterization, we vary the number of features used in the optimization. Specifically, we solve (3) with $\Delta = [k]$ and vary the number of features k from 0 to p. Here we consider the *train→prune* algorithm, where we first solve for $\hat{\boldsymbol{\beta}}([k])$ and obtain our pruned model $\hat{\boldsymbol{\beta}}_s^P([k])$ by applying magnitude, Hessian or Oracle pruning



(a) Identity covariance, spiked latent weights.

(b) Spiked covariance, identical latent weights.

Figure 3: Theoretical predictions for various pruning strategies in linear models with $s/p = 0.1$ and $n/p = 0.3$.

(cf. $P \in \{M, H, O\}$). Retraining curves are omitted here, but they can be found in Fig. 7 of (Chang et al. 2020). Since $\boldsymbol{\lambda}$ is non-increasing, the indices are sorted by saliency score; thus, Oracle pruning always picks the first s indices. Solid lines represent analytic predictions, while markers are empirical results. The vertical dashed line is the sparsity level s/p. The horizontal dashed line highlights the minimum risk among all underparameterized solutions ($k \leq n$) and all solutions obtained by a final retraining.

In Fig. 3a, we set $\boldsymbol{\Sigma} = \boldsymbol{I}_p$ and $\boldsymbol{\beta}^\star = \sqrt{\boldsymbol{\lambda}}$. Note, that the analytic curves correctly predict the test risk and the double descent behavior. Observe that the Hessian and Magnitude pruning coincide here, since the diagonal of the empirical covariance is essentially identity. In contrast, Fig. 3b emphasizes the role of the feature covariance by setting $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\lambda})$ and $\boldsymbol{\beta}^\star$ to be the all ones vector. In this scenario, we observe that Hessian pruning performs better compared to Fig. 3a and also outperforms Magnitude pruning. This is because the empirical covariance helps distinguish the salient indices. Importantly, for Hessian and Oracle pruning, the optimal sparse model is achieved in the highly overparameterized regime $k = p$. Notably, the achieved performance at $k = p$ is strictly better than the horizontal dashed line, which highlights the optimal risk among all underparameterized solutions $k \leq n$ and all retraining solutions (see also (Chang et al. 2020) Sec. A). This has two striking consequences. First, *retraining can in fact hurt the performance*; because the *train→prune* performance at $k = p$ is strictly better than *train→prune→retrain* for all k. Second, *overparameterized pruning can be provably better than solving the sparse model with the knowledge of the most salient features* as $k = p$ is also strictly better than $k = s$.

3.2 Random Features Regression

We relate an ERM problem (3) with nonlinear map ϕ to an equivalent LGP. This will allow us to use our theoretical results about the latter to characterize the properties of the original nonlinear map. We ensure the equivalence by properly setting up the LGP to exhibit similar second order statistics as the original problem.

Definition 2 (Equivalent Linear Problem) *Given distribution $(\boldsymbol{x}, y) \sim \mathcal{D}$, the equivalent LGP($\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma$) with n samples is given with parameters $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$,*

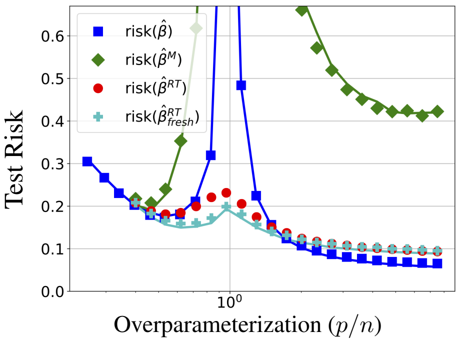


Figure 4: Illustration of the slight mismatch between standard retraining (with same samples, red markers) and retraining with fresh samples (cyan markers/line).

$\boldsymbol{\beta}^\star = \boldsymbol{\Sigma}^{-1}\,\mathbb{E}[y\boldsymbol{x}]$ and $\sigma = \mathbb{E}[(y - \boldsymbol{x}^T\boldsymbol{\beta}^\star)^2]^{1/2}$.

In Section 5, we formalize the DC of LGPs, which enables us to characterize pruning/retraining dynamics. Then, we empirically verify that DC and pruning dynamics of equivalent LGPs can successfully predict the original problem (3) with non-linear features. The idea of setting up and studying equivalent LGPs as a proxy to nonlinear models, has been recently used in the emerging literature of high-dimensional learning, for predicting the performance of the original ERM task (Montanari et al. 2019; Goldt et al. 2020; Abbasi, Salehi, and Hassibi 2019; Dereziński, Liang, and Mahoney 2019). This work goes beyond prior art, which focuses on ERM, by demonstrating that we can also successfully predict the pruning/retraining dynamics. Formalizing the performance equivalence between LGP and equivalent problem is an important future research avenue and it can presumably be accomplished by building on the recent high-dimensional universality results such as (Oymak and Tropp 2018; Hu and Lu 2020; Abbasi, Salehi, and Hassibi 2019; Goldt et al. 2020).

In Fig. 2, we study random feature regression to approximate a synthetic nonlinear distribution. Specifically, data has the following distribution: Given input $\boldsymbol{a} \sim \mathcal{N}(0, \boldsymbol{I}_d)$, we generate random unit norm $\boldsymbol{\beta}^1 \in \mathbb{R}^d, \boldsymbol{\beta}^2 \in \mathbb{R}^d$ and set the label to be a quadratic function given by $y = \boldsymbol{a}^T\boldsymbol{\beta}^1 + (\boldsymbol{a}^T\boldsymbol{\beta}^2)^2$. Then, we fix $\boldsymbol{R} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ and we generate ReLU features $\boldsymbol{x} = \text{ReLU}(\boldsymbol{R}\boldsymbol{a})$, where $\boldsymbol{R}$ corresponds to the input layer of a two-layer network. The markers in Fig. 2 are obtained by solving RFR and pruning and retraining with varying sparsity targets ($s, 2s, 4s$ with $s/n = 10\%$). Here, $d = 10, n = 200$. For each marker, the results are averages of 50 $\boldsymbol{R} \in \mathbb{R}^{p \times d}$ realizations and 10 iterations for each choice of $\boldsymbol{R}$. The lines are obtained via our DC of the equivalent LGP (by using Defs. 4 and 5) where the latent parameter $\boldsymbol{\beta}^\star$, noise σ and the covariance $\boldsymbol{\Sigma}$ of the RFR problem are calculated for fixed realization of the input layer $\boldsymbol{R}$ (similarly averaged over 50 random $\boldsymbol{R}$). The blue line is the performance of usual RFR with growing number of features p. The other lines are obtained by solving RFR with p features, pruning and retraining the solution to fixed sparsity levels ($s, 2s, 4s$) with $s/n = 0.1$. Importantly, the risks of the retrained models exhibit double descent and are minimized when $p \gg n$ despite fixed model size / sparsity. Our theory and empirical curves exhibit a good match. (The slight mismatch of the red curve/markers is explained in Fig. 4.) The results demonstrate the importance of overparameterization for RF pruning, which corresponds to picking *random features smartly*. Here, the coefficients of least-squares act like a scoring function for the saliency of random features and capture how well they are aligned with the target function. The fact that the risk of the pruned models is minimized in the overparameterized regime implies that least-squares regression succeeds in properly selecting salient random features from a larger candidate set. In the context of deep learning, our discussion can be interpreted as *pruning hidden nodes of the network*.

Predicting retraining performance. As discussed in Sec. 5 and Def. 5, for the retraining stage, our DC is accomplished by assuming that retraining phase uses n fresh training examples (i.e. a new dataset $\mathcal{S}_{\text{fresh}}$). Let us denote the resulting model by $\hat{\boldsymbol{\beta}}^{RT}_{\text{fresh}}$. In Fig. 4, we use exactly the same setting as in Fig. 2, but only show the case of sparsity $4s$ for which the mismatch is observed. Observe that our analytical predictions accurately capture the risk of retraining with fresh samples. However, we observe a discrepancy with the true risk of retraining (without fresh samples) around the interpolation threshold. Also shown the risk of the original ERM solution before pruning (in blue) and of the magnitude-pruned model (before any retraining). Perhaps surprisingly, Fig. 2 shows that this DC correctly captures the performance of $\hat{\boldsymbol{\beta}}^{RT}$ with the exception of the red curve ($4s$). Fig. 4 focuses on this instance and shows that our DC indeed perfectly predicts the fresh retraining performance and verifies the slight empirical mismatch between $\hat{\boldsymbol{\beta}}^{RT}$ and $\hat{\boldsymbol{\beta}}^{RT}_{\text{fresh}}$.

3.3 Neural Network Experiments

Finally, we study pruning deep neural networks. Inspired by (Nakkiran et al. 2019), we train ResNet-20 with changeable filters over CIFAR-10. Here, the filter number k is equivalent to the width of the model and it controls the overall model size. As the width of ResNet-20 changes, the fitting performance of the dataset varies. Here, we apply *train→prune→retrain*. Select s as the sparsity target and s-filter ResNet-20 model as the base model with N_s parameters. First, we train a dense model with k filters and N_k parameters, $N_k \gg N_s$, and prune it by only keeping the largest N_s entries in absolute value non-zero. N_k grows approximately quadratically in k. Now, the sparse model shares the same number of parameters amenable to training as the base model. Finally, we retain the pruned network and record its performance on the same dataset and same configuration. In Fig. 1, we plot the training (dashed) and test (solid) errors of dense and sparse models. All the neural experiments are trained with Adam optimization and 0.001 learning rate for 1000 epochs, with data augmentation. The blue line corresponds to training of a dense model with width-k. Green, yellow and red lines correspond to sparsity targets $s \in \{5, 8, 10\}$, with around 28,000, 70,000 and 109,000 trainable parameters, for which a dense model of width-k is first pruned to achieve the exact same number of nonzeros as a dense model of width-s and then retrained over the identified nonzero pattern. Surprisingly, all curves interpolate (achieve zero training error) around the same width parameter despite varying sparsity. As the width k grows, the training and test errors decrease for all 5-, 8-, 10-filter base models, except for the shaded double descent range and the best test error is always achieved in the overparameterized regime (large width). These experiments again verify the main insight revealed to us by studying simpler linear and random-feature models, that is, training a larger model, followed by appropriate pruning, can preform better than training a small model from the beginning. Another worth-mentioning observation is that with appropriate sparsity level (here, 10) the pruned model has prediction performance comparable to the dense model. Finally and interestingly, the test error dynamics of the pruned model exhibit a double descent that resembles that of the dense model (previously observed in (Nakkiran et al. 2019)).

3.4 Further Intuitions on The Denoising Effect of Overparameterization

To provide further insights into the pruning benefits of overparameterization, consider a simple linear model (as in Def 1) with $n \geq p \geq s$, noise level $\sigma = 0$ and identity covariance. Suppose our goal is estimating the coefficients $\boldsymbol{\beta}^\star_\Delta$ for some fixed index set $\Delta \subset [p]$ with $|\Delta| = s$. For pruning, we can pick Δ to be the most salient/largest entries. If we solve the smaller regression problem over Δ, $\hat{\boldsymbol{\beta}}(\Delta)$ will only provide a noisy estimate of $\boldsymbol{\beta}^\star_\Delta$. The reason is that, the signal energy of the missing features $[p] - \Delta$ acts as a noise uncorrelated with the features in Δ. Conversely, if we solve ERM with all features (the larger problem), we perfectly recover $\boldsymbol{\beta}^\star$ due to zero noise and invertibility ($n \geq p$). Then one can also perfectly estimate $\boldsymbol{\beta}^\star_\Delta$. This simple argument, which is partly inspired by the missing feature setup in (Hastie et al. 2019), shows that solving the larger problem with more parameters can have a "denoising-like effect" and perform better than the small problem. Our contribution obviously goes well beyond this discussion and theoretically characterizes the exact asymptotics, handles the general covariance model and all (n, p, s) regimes, and also highlights the importance of the overparameterized regime $n \ll p$.

4 Understanding the Benefits of Retraining

On the one hand, the study of LGPs in Fig. 3 and Fig. 7 of (Chang et al. 2020) suggest that retraining can actually hurt the performance. On the other hand, in practice and in the RFR experiments of Fig. 4, retraining is crucial; compare the green $\hat{\boldsymbol{\beta}}^M$ and red $\hat{\boldsymbol{\beta}}^{RT}$ curves and see (Chang et al. 2020) Section A for further DNN experiments. Here, we argue that the benefit of retraining is connected to the correlations between input features. Indeed, the covariance/Hessian matrices associated with RF and DNN regression are not diagonal (as was the case in Fig. 3). To build intuition, imagine that only a single feature suffices to explain the label. If there are multiple other features that can similarly explain the label, the model prediction will be shared across these features. Then, pruning will lead to a biased estimate, which can be mitigated by retraining. The following lemma formalizes this intuition under an instructive setup, where the features are perfectly correlated.

Lemma 1 *Suppose $\mathcal{S}$ is drawn from an LGP$(\sigma, \boldsymbol{\Sigma}, \boldsymbol{\beta}^\star)$ as in Def. 1 where rank$(\boldsymbol{\Sigma}) = 1$ with $\boldsymbol{\Sigma} = \boldsymbol{\lambda}\boldsymbol{\lambda}^T$ for $\boldsymbol{\lambda} \in \mathbb{R}^p$. Define $\zeta = \mathbb{T}_s(\boldsymbol{\lambda})^2/\|\boldsymbol{\lambda}\|_{\ell_2}^2$. For magnitude and Hessian pruning ($P \in \{M, H\}$) and the associated retraining, we have the following excess risks with respect to $\boldsymbol{\beta}^\star$*

$$\mathbb{E}_{\mathcal{S}}[\mathcal{L}(\hat{\boldsymbol{\beta}}_s^P)] - \mathcal{L}(\boldsymbol{\beta}^\star) = \frac{\zeta^2\sigma^2}{n-2} + \underbrace{(1-\zeta)^2(\boldsymbol{\lambda}^T\boldsymbol{\beta}^\star)^2}_{\textit{Error due to bias}} \tag{5}$$

$$\mathbb{E}_{\mathcal{S}}[\mathcal{L}(\hat{\boldsymbol{\beta}}_s^{RT})] - \mathcal{L}(\boldsymbol{\beta}^\star) = \sigma^2/(n-2). \tag{6}$$

The lemma reveals that pruning the model leads to a biased estimator of the label. Specifically, the bias coefficient $1-\zeta$ arises from the missing predictions of the pruned features (which correspond to the small coefficients of $|\boldsymbol{\lambda}|$). In contrast, regardless of s, retraining always results in an unbiased estimator with the exact same risk as the dense model which quickly decays in sample size n. The reason is that retraining enables the remaining features to account for the missing predictions. Here, this is accomplished perfectly, due to the fully correlated nature of the problem. In particular, this is in contrast to the diagonal covariance (Fig. 3), where the missing features act like uncorrelated noise during retraining.

5 Main Results

Here, we present our main theoretical result: a sharp asymptotic characterization of the distribution of the solution to overparameterized least-squares for correlated designs. We further show how this leads to a sharp prediction of the risk of magnitude-based pruning. Concretely, for the rest of this section, we assume the linear Gaussian problem (LGP) of Definition 1, the overparameterized regime $k = p > n$ and the min-norm model

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_{\ell_2} \text{ s.t. } \boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}. \tag{7}$$

As mentioned in Sec. 2, $\hat{\boldsymbol{\beta}}$ is actually given in closed-form as $\hat{\boldsymbol{\beta}} = \boldsymbol{X}^\dagger\boldsymbol{y}$. Interestingly, our analysis of the distribution of $\hat{\boldsymbol{\beta}}$ does not rely on the closed-form expression, but rather follows by viewing $\hat{\boldsymbol{\beta}}$ as the solution to the convex linearly-constrained quadratic program in (7). Specifically, our analysis uses the framework of the convex Gaussian min-max Theorem (CGMT) (Thrampoulidis, Oymak, and Hassibi 2015), which allows to study rather general inference optimization problems such as the one in (7), by relating them with an auxiliary optimization that is simpler to analyze (Stojnic 2013; Oymak, Thrampoulidis, and Hassibi 2013; Thrampoulidis, Oymak, and Hassibi 2015; Thrampoulidis, Abbasi, and Hassibi 2018; Salehi, Abbasi, and Hassibi 2019; Taheri, Pedarsani, and Thrampoulidis 2020). Due to space considerations, we focus here on the more challenging overparameterized regime and defer the analysis of the underparameterized regime to (Chang et al. 2020).

5.1 Distributional Characterization of the Overparameterized Linear Gaussian Models

Notation: We first introduce additional notation necessary to state our theoretical results. $\odot$ denotes the entrywise product of two vectors and $\mathbf{1}_p$ is the all ones vector in $\mathbb{R}^p$. The *empirical distribution* of a vector $\boldsymbol{x} \in \mathbb{R}^p$ is given by $\frac{1}{p}\sum_{i=1}^p \delta_{x_i}$, where δ_{x_i} denotes a Dirac delta mass on x_i. Similarly, the empirical joint distribution of vectors $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^p$ is $\frac{1}{p}\sum_{i=1}^p \delta_{(x_i, x_i')}$. The *Wasserstein-k* ($W_k$) distance between two measures μ and ν is defined as $W_k(\mu,\nu) \equiv \left(\inf_\rho \mathbb{E}_{(X,Y)\sim\rho}|X-Y|^k\right)^{1/k}$, where the infimum is over all the couplings of μ and ν, i.e. all random variables (X, Y) such that $X \sim \mu$ and $Y \sim \nu$ marginally. A sequence of probability distributions ν_p on $\mathbb{R}^m$ converges in W_k to ν, written $\nu_p \overset{W_k}{\Longrightarrow} \nu$, if $W_k(\nu_p, \nu) \to 0$ as $p \to \infty$. Finally, we say that a function $f: \mathbb{R}^m \to \mathbb{R}$ is pseudo-Lipschitz of order k, denoted $f \in$ PL(k), if there is a constant $L > 0$ such that for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$, $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq L(1 + \|\boldsymbol{x}\|_{\ell_2}^{k-1} + \|\boldsymbol{y}\|_{\ell_2}^{k-1})\|\boldsymbol{x}-\boldsymbol{y}\|_2$. We call L the PL(k) constant of f. An equivalent definition of W_k convergence is that, for any $f \in$ PL(k), $\lim_{p\to\infty}\mathbb{E} f(X_p) = \mathbb{E} f(X)$, where expectation is with respect to $X_p \sim \nu_p$ and $X \sim \nu$. For a sequence of random variables $\mathcal{X}_p$ that converge in probability to some constant c in the limit of Assumption 1 below, we write $\mathcal{X}_p \xrightarrow{P} c$.

Next, we formalize the set of assumption under which our analysis applies. Our asymptotic results hold in the linear asymptotic regime specified below.

Assumption 1 *We focus on a double asymptotic regime where $n, p, s \to \infty$ at fixed overparameterization ratio $\kappa := p/n > 0$ and sparsity level $\alpha := s/p \in (0, 1)$.*

Additionally, we require certain mild assumptions on the behavior of the covariance matrix $\boldsymbol{\Sigma}$ and of the true latent vector $\boldsymbol{\beta}^\star$. For simplicity, we assume here that $\boldsymbol{\Sigma} = \text{diag}([\boldsymbol{\Sigma}_{1,1}, \ldots, \boldsymbol{\Sigma}_{p,p}])$.

Assumption 2 *The covariance matrix $\boldsymbol{\Sigma}$ is diagonal and there exist constants $\Sigma_{\min}, \Sigma_{\max} \in (0, \infty)$ such that: $\Sigma_{\min} \leq \boldsymbol{\Sigma}_{i,i} \leq \Sigma_{\max}$, for all $i \in [p]$.*

Assumption 3 *The joint empirical distribution of $\{(\boldsymbol{\Sigma}_{i,i}, \sqrt{p}\boldsymbol{\beta}_i^\star)\}_{i\in[p]}$ converges in Wasserstein-k distance to a probability distribution μ on $\mathbb{R}_{>0} \times \mathbb{R}$ for some $k \geq 4$. That is $\frac{1}{p}\sum_{i\in[p]} \delta_{(\boldsymbol{\Sigma}_{i,i}, \sqrt{p}\boldsymbol{\beta}_i^\star)} \overset{W_k}{\Longrightarrow} \mu$.*

With these, we are ready to define, what will turn out to be, the asymptotic DC in the overparameterized regime.

Definition 3 (Asymptotic DC – Overparameterized regime) *Let random variables $(\Lambda, B) \sim \mu$ (where μ is defined in Assumption 3) and fix $\kappa > 1$. Define parameter ξ as the unique positive solution to the following equation*

$$\mathbb{E}_\mu\left[\left(1 + (\xi\cdot\Lambda)^{-1}\right)^{-1}\right] = \kappa^{-1}. \tag{8}$$

Further define the positive parameter γ as follows:

$$\gamma := \left(\sigma^2 + \mathbb{E}_\mu\left[\frac{B^2\Lambda}{(1+\xi\Lambda)^2}\right]\right)\Big/\left(1 - \kappa\,\mathbb{E}_\mu\left[\frac{1}{(1+(\xi\Lambda)^{-1})^2}\right]\right). \tag{9}$$

With these and $H \sim \mathcal{N}(0, 1)$, define the random variable

$$X_{\kappa,\sigma^2}(\Lambda, B, H) := \left(1 - \frac{1}{1+\xi\Lambda}\right)B + \sqrt{\kappa}\frac{\sqrt{\gamma}\,\Lambda^{-1/2}}{1+(\xi\Lambda)^{-1}}H,$$

and let Π_{κ,σ^2} be its distribution.

Our main result establishes asymptotic convergence of the empirical distribution of $(\sqrt{p}\hat{\boldsymbol{\beta}}, \sqrt{p}\boldsymbol{\beta}^\star, \boldsymbol{\Sigma})$ for a rich class of test functions. These are the functions within $\mathrm{PL}(3)$ that become $\mathrm{PL}(2)$ when restricted to the first two indices. Formally, we define this class of functions as follows

$$\begin{aligned}\mathcal{F} := \{f : \mathbb{R}^2 \times \mathcal{Z} \to \mathbb{R},\ f \in \mathrm{PL}(3) \text{ and}\\ \sup_{z\in\mathcal{Z}} \text{“PL(2) constant of f}(\cdot,\cdot,\text{z})\text{”} < \infty\}.\end{aligned} \tag{10}$$

For pruning analysis, we set $\mathcal{Z} = [\Sigma_{\min}, \Sigma_{\max}]$ and define

$$\begin{aligned}\mathcal{F}_{\mathcal{L}} := \{f : \mathbb{R}^2 \times \mathcal{Z} \to \mathbb{R} \mid f(x,y,z) = z(y - g(x))^2\\ \text{where } g(\cdot) \text{ is Lipschitz}\}.\end{aligned} \tag{11}$$

As discussed below, $\mathcal{F}_{\mathcal{L}}$ is important for predicting the risk of the (pruned) model. In (Chang et al. 2020), we prove that $\mathcal{F}_{\mathcal{L}} \subset \mathcal{F}$. We are now ready to state our main theoretical result.

Theorem 1 (Asymptotic DC – Overparameterized LGP) *Fix $\kappa > 1$ and suppose Assumptions 2 and 3 hold. Recall the solution $\hat{\boldsymbol{\beta}}$ from* (7) *and let $X_{\kappa,\sigma^2} \sim \Pi_{\kappa,\sigma^2}$ as in Definition 3. Let $f : \mathbb{R}^3 \to \mathbb{R}$ be a function in $\mathcal{F}$ defined in* (10). *We have that*

$$\frac{1}{p}\sum_{i=1}^{p} f(\sqrt{p}\hat{\boldsymbol{\beta}}_i, \sqrt{p}\boldsymbol{\beta}^\star_i, \boldsymbol{\Sigma}_{i,i}) \xrightarrow{P} \mathbb{E}\left[f(X_{\kappa,\sigma^2}, B, \Lambda)\right]. \tag{12}$$

As advertised, Theorem 1 fully characterizes the joint empirical distribution of the min-norm solution, the latent vector and the covariance spectrum. The asymptotic DC allows us to precisely characterize several quantities of interest, such as estimation error, generalization error etc.. For example, a direct application of (12) to the function $f(x,y,z) = z(y-x)^2 \in \mathcal{F}_{\mathcal{L}} \subset \mathcal{F}$ directly yields the risk prediction of the min-norm solution recovering (Hastie et al. 2019, Thm. 3) as a special case. Later in this section, we show how to use Theorem 1 towards the more challenging task of precisely characterizing the risk of magnitude-based pruning.

Before that, let us quickly remark on the technical novelty of the theorem. Prior work has mostly applied the CGMT to isotropic features. Out of these, only very few obtain DC, (Thrampoulidis, Xu, and Hassibi 2018; Miolane and Montanari 2018), while the majority focuses on simpler metrics, such as squared-error. Instead, Theorem 1 considers correlated designs and the overparameterized regime. The most closely related work in that respect is (Montanari et al. 2019), which very recently obtained the DC of the max-margin classifier. Similar to us, they use the CGMT, but their analysis of the auxiliary optimization is technically different to ours. Our approach is similar to (Thrampoulidis, Xu, and Hassibi 2018), but extra technical effort is needed to account for correlated designs and the overparameterized regime.

5.2 From DC to Risk Characterization

First, we consider a simpler "threshold-based" pruning method that applies a fixed threshold at every entry of $\hat{\boldsymbol{\beta}}$. Next, we relate this to magnitude-based pruning and obtain a characterization for the performance of the latter. In order to define the threshold-based pruning vector, let

$$\mathcal{T}_t(x) = \begin{cases} x & \text{if } |x| > t \\ 0 & \text{otherwise} \end{cases},$$

be the hard-thresholding function with fixed threshold $t \in \mathbb{R}_+$. Define $\hat{\boldsymbol{\beta}}^{\mathcal{T}}_t := \mathcal{T}_{t/\sqrt{p}}(\hat{\boldsymbol{\beta}})$, where $\mathcal{T}_t$ acts componentwise. Then, the population risk of $\hat{\boldsymbol{\beta}}^{\mathcal{T}}_t$ becomes

$$\begin{aligned}\mathcal{L}(\hat{\boldsymbol{\beta}}^{\mathcal{T}}_t) &= \mathbb{E}_{\mathcal{D}}[(\boldsymbol{x}^T(\boldsymbol{\beta}^\star - \hat{\boldsymbol{\beta}}^{\mathcal{T}}_t) + \sigma z)^2]\\ &= \sigma^2 + \frac{1}{p}\sum_{i=1}^{p} \boldsymbol{\Sigma}_{i,i}\big(\sqrt{p}\boldsymbol{\beta}^\star_i - \mathcal{T}_t(\sqrt{p}\hat{\boldsymbol{\beta}}_i)\big)^2\\ &\xrightarrow{P} \sigma^2 + \mathbb{E}\left[\Lambda\left(B - \mathcal{T}_t(X_{\kappa,\sigma^2})\right)^2\right].\end{aligned} \tag{13}$$

In the second line above, we note that $\sqrt{p}\mathcal{T}_{t'}(x) = \mathcal{T}_{\sqrt{p}t'}(\sqrt{p}x)$. In the last line, we apply (12), after recognizing that the function $(x,y,z) \mapsto z(y - \mathcal{T}_t(x))^2$ is a member of the $\mathcal{F}_{\mathcal{L}}$ family defined in (11). As in (12), the expectation here is with respect to $(\Lambda, B, H) \sim \mu \otimes \mathcal{N}(0,1)$.

Now, we show how to use (13) and Theorem 1 to characterize the risk of the magnitude-based pruned vector $\boldsymbol{\beta}^M_s := \mathbb{T}_s(\hat{\boldsymbol{\beta}})$. Recall, here from Assumption 1 that $s = \alpha p$. To relate $\hat{\boldsymbol{\beta}}^M_s$ to $\hat{\boldsymbol{\beta}}^{\mathcal{T}}_t$, consider the set $\mathcal{S}_t := \{i \in [p] : \sqrt{p}|\hat{\boldsymbol{\beta}}_i| \geq t\}$ for some constant $t \in \mathbb{R}_+$ (not scaling with n, p, s). Note that the ratio $|\mathcal{S}_t|/p$ is equal to

$$p^{-1}\sum_{i=1}^{p} \mathbb{1}_{[\sqrt{p}|\hat{\boldsymbol{\beta}}_i| \geq t]} \xrightarrow{P} \mathbb{E}[\mathbb{1}_{[|X_{\kappa,\sigma^2}| \geq t]}] = \mathbb{P}\left(|X_{\kappa,\sigma^2}| \geq t\right).$$

Here, $\mathbb{1}$ denotes the indicator function and the convergence follows from Theorem 1 when applied to a sequence of bounded Lipschitz functions approximating the indicator. Thus, by choosing

$$t^\star := \sup\left\{t \in \mathbb{R} : \mathbb{P}(|X_{\kappa,\sigma^2}| \geq t) \geq \alpha\right\}, \tag{14}$$

it holds that $|\mathcal{S}_t|/p \xrightarrow{P} \alpha$. In words, and observing that X_{κ,σ^2} admits a continuous density (due to the Gaussian variable H): for any $\varepsilon > 0$, in the limit of $n, p, s \to \infty$, the vector $\hat{\boldsymbol{\beta}}^{\mathcal{T}}_{t^\star}$ has $(1 \pm \varepsilon)\alpha p = (1 \pm \varepsilon)s$ non-zero entries, which correspond to the largest magnitude entries of $\hat{\boldsymbol{\beta}}$, with probability approaching1. Since this holds for arbitrarily small $\varepsilon > 0$, recalling $t^\star$ as in (14), we can conclude from (13) that the risk of the magnitude-pruned model converges as follows.

Corollary 1 (Risk of Magnitude-pruning) *Let the same assumptions and notation as in the statement of Theorem 1 hold. Specifically, let $\hat{\boldsymbol{\beta}}$ be the min-norm solution in* (7) *and $\hat{\boldsymbol{\beta}}^M_s := \mathbb{T}_s(\boldsymbol{\beta})$ the magnitude-pruned model at sparsity s. Recall the threshold $t^\star$ from* (14). *The risk of $\hat{\boldsymbol{\beta}}^M_s$ satisfies the following in the limit of $n, p, s \to \infty$ at rates $\kappa := p/n > 1$ and $\alpha := s/p \in (0,1)$ (cf. Assumption 1):*

$$\mathcal{L}(\hat{\boldsymbol{\beta}}^M_s) \xrightarrow{P} \sigma^2 + \mathbb{E}\left[\Lambda\left(B - \mathcal{T}_{t^\star}(X_{\kappa,\sigma^2})\right)^2\right],$$

where the expectation is over $(\Lambda, B, H) \sim \mu \otimes \mathcal{N}(0,1)$.

5.3 Non-asymptotic DC and Retraining Formula

While Theorem 1 is stated in the asymptotic regime, during analysis, the DC arises in a non-asymptotic fashion. The following definition is the non-asymptotic counterpart of Def. 3. We remark that this definition applies to arbitrary covariance (not necessarily diagonal) by applying a simple eigen-rotation before and after the DC formula associated with the diagonalized covariance.

Definition 4 (Non-asymptotic DC) *Fix $p > n \geq 1$ and set $\kappa = p/n > 1$. Given $\sigma > 0$, covariance $\boldsymbol{\Sigma} = \boldsymbol{U}\mathit{diag}(\boldsymbol{\lambda})\boldsymbol{U}^T$ and latent vector $\boldsymbol{\beta}$, set $\bar{\boldsymbol{\beta}} = \boldsymbol{U}^T\boldsymbol{\beta}$ and define the unique non-negative terms $\xi, \gamma, \boldsymbol{\zeta} \in \mathbb{R}^p$ and $\boldsymbol{\phi} \in \mathbb{R}^p$ as follows:*

$$\xi > 0 \quad \textit{is the solution of} \quad \kappa^{-1} = p^{-1}\sum_{i=1}^{p}\left(1 + (\xi\boldsymbol{\lambda}_i)^{-1}\right)^{-1},$$

$$\gamma = \frac{\sigma^2 + \sum_{i=1}^{p}\boldsymbol{\lambda}_i\boldsymbol{\zeta}_i^2\bar{\boldsymbol{\beta}}_i^2}{1 - \frac{\kappa}{p}\sum_{i=1}^{p}(1 + (\xi\boldsymbol{\lambda}_i)^{-1})^{-2}},$$

$$\boldsymbol{\zeta}_i = (1 + \xi\boldsymbol{\lambda}_i)^{-1} \quad , \quad \boldsymbol{\phi}_i = \kappa\gamma(1 + (\xi\boldsymbol{\lambda}_i)^{-1})^{-2},\ 1 \leq i \leq p.$$

The non-asymptotic distributional prediction is given by the following $\boldsymbol{U}$-rotated normal distribution

$$\mathcal{D}_{\sigma,\boldsymbol{\Sigma},\boldsymbol{\beta}} = \boldsymbol{U}\mathcal{N}((\mathbf{1}_p - \boldsymbol{\zeta}) \odot \bar{\boldsymbol{\beta}}, p^{-1}\mathit{diag}(\boldsymbol{\lambda}^{-1} \odot \boldsymbol{\phi})).$$

We remark that this definition is similar in spirit to the concurrent/recent work (Li et al. 2020). However, unlike this work, here we prove the asymptotic correctness of the DC, we use it to rigorously predict the pruning performance and also extend this to retraining DC as discussed next.

Retraining DC. As the next step, we would like to characterize the DC of the solution after retraining, i.e., $\hat{\boldsymbol{\beta}}^{RT}$. We carry out the retraining derivation (for magnitude pruning) as follows. Let $\mathcal{I} \subset [p]$ be the nonzero support of the pruned vector $\hat{\boldsymbol{\beta}}_s^M$. Re-solving (3) restricted to the features over $\mathcal{I}$ corresponds to a linear problem with effective feature covariance $\boldsymbol{\Sigma}_{\mathcal{I}}$ with support of non-zeros restricted to $\mathcal{I} \times \mathcal{I}$. For this feature covariance, we can also calculate the effective noise level and global minima of the population risk $\boldsymbol{\beta}_{\mathcal{I}}^\star$. The latter has the closed-form solution $\boldsymbol{\beta}_{\mathcal{I}}^\star = \boldsymbol{\Sigma}_{\mathcal{I}}^\dagger\boldsymbol{\Sigma}\boldsymbol{\beta}^\star$. The effective noise is given by accounting for the risk change due to the missing features via $\sigma_{\mathcal{I}} = (\sigma^2 + \boldsymbol{\beta}^{\star T}\boldsymbol{\Sigma}\boldsymbol{\beta}^\star - \boldsymbol{\beta}_{\mathcal{I}}^{\star T}\boldsymbol{\Sigma}_{\mathcal{I}}\boldsymbol{\beta}_{\mathcal{I}}^\star)^{1/2}$. With these terms in place, fixing $\mathcal{I}$ and using Def. 4, the retraining prediction becomes $\mathcal{D}_{\sigma_{\mathcal{I}},\boldsymbol{\Sigma}_{\mathcal{I}},\boldsymbol{\beta}_{\mathcal{I}}^\star}$. This process is summarized below.

Definition 5 (Retraining DC) *Consider the setting of Def. 4 with $\sigma, \boldsymbol{\Sigma}, \boldsymbol{\beta}^\star$ and sparsity target s. The sample $\hat{\boldsymbol{\beta}}^{RT}$ from the retraining distribution $\mathcal{D}_{\sigma,\boldsymbol{\Sigma},\boldsymbol{\beta}^\star}^{RT,s}$ is constructed as follows. Sample $\hat{\boldsymbol{\beta}} \sim \mathcal{D}_{\sigma,\boldsymbol{\Sigma},\boldsymbol{\beta}^\star}$ and compute the set of the top-s indices $\mathcal{I} = \mathcal{I}(\mathbb{T}_s(\hat{\boldsymbol{\beta}}))$. Given $\mathcal{I}$, obtain the effective covariance $\boldsymbol{\Sigma}_{\mathcal{I}} \in \mathbb{R}^{p\times p}$, population minima $\boldsymbol{\beta}_{\mathcal{I}}^\star \in \mathbb{R}^p$, and the noise level $\sigma_{\mathcal{I}} > 0$ as described above. Draw $\hat{\boldsymbol{\beta}}^{RT} \sim \mathcal{D}_{\sigma_{\mathcal{I}},\boldsymbol{\Sigma}_{\mathcal{I}},\boldsymbol{\beta}_{\mathcal{I}}^\star}$.*

Observe that, the support $\mathcal{I}$ depends on the samples $\mathcal{S}$ via $\hat{\boldsymbol{\beta}}$. Thus, our retraining DC is actually derived for the scenario when the retraining phase uses a fresh set of n samples to break the dependence between $\mathcal{I}, \mathcal{S}$ (which obtains $\hat{\boldsymbol{\beta}}_{\text{fresh}}^{RT}$). Despite this, we empirically observe that the retraining DC predicts the regular retraining (reusing $\mathcal{S}$) performance remarkably well and perfectly predicts $\hat{\boldsymbol{\beta}}_{\text{fresh}}^{RT}$ as discussed in Figs. 2 and 4. Finally, we defer the formalization of the retraining analysis to a future work. This includes proving that $\hat{\boldsymbol{\beta}}_{\text{fresh}}^{RT}$ obeys Def. 5 asymptotically as well as directly studying $\hat{\boldsymbol{\beta}}^{RT}$ by capturing the impact of the $\mathcal{I}, \mathcal{S}$ dependency.

6 Conclusions and Future Directions

This paper sheds light on under-explored phenomena in pruning practices for neural network model compression. On a theoretical level, we prove an accurate distributional characterization (DC) for the solution of overparameterized least-squares for linear models with correlated Gaussian features. Our DC allows to precisely characterize the pruning performance of popular pruning methods, such as magnitude pruning. Importantly, our DC combined with a linear Gaussian equivalence, leads to precise analytic formulas for the pruning performance of nonlinear random feature models. On the experimental side, we provide a thorough study of overparameterization and pruning with experiments on linear models, random features and neural nets with growing complexity. Our experiments reveal striking phenomena such as a novel double descent behavior for model pruning and the power of overparameterization. They also shed light on common practices such as retraining after pruning.

Going forward, there are several exciting directions to pursue. First, it would be insightful to study whether same phenomena occur for other loss functions in particular for cross-entropy. Second, this work focuses on unregularized regression tasks and it is important to identify optimal regularization schemes for pruning purposes. For instance, should we use classical ℓ_1/ℓ_2 regularization or can we refine them by injecting problem priors such as covariance information? Finally, going beyond pruning, using DC, one can investigate other compression techniques that process the output of the initial overparameterized learning problem, such as model quantization and distillation.

Acknowledgments

S. Oymak is partially supported by the NSF award CNS-1932254 and by the NSF CAREER award CCF-2046816. C. Thrampoulidis was partially supported by the NSF under Grant Numbers CCF-2009030.

Ethical Impact

While deep learning is transformative in wide swath of applications, it comes at a cost: State-of-the-art deep learning models tend to be very large and consume significant energy during inference. The race for larger and better models and growing list of applications exacerbates this carbon footprint problem. Thus there is an urgent need for better and more principled model compression methods to help build environmentally friendly ML models. This work responds to this need by establishing the fundamental algorithmic principles and guarantees behind the contemporary model compression algorithms and by shedding light on the design of lightweight energy- and compute-efficient neural networks. We do not see an ethical concern associated with this work.

References

Abbasi, E.; Salehi, F.; and Hassibi, B. 2019. Universality in learning from linear measurements. In *Advances in Neural Information Processing Systems*, 12372–12382.

Arora, S.; Cohen, N.; and Hazan, E. 2018. On the optimization of deep networks: Implicit acceleration by overparameterization. In *35th International Conference on Machine Learning*.

Bartlett, P. L.; Long, P. M.; Lugosi, G.; and Tsigler, A. 2020. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences* .

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

Belkin, M.; Hsu, D.; and Xu, J. 2019. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571* .

Belkin, M.; Ma, S.; and Mandal, S. 2018. To Understand Deep Learning We Need to Understand Kernel Learning. In *International Conference on Machine Learning*, 541–549.

Belkin, M.; Rakhlin, A.; and Tsybakov, A. B. 2019. Does data interpolation contradict statistical optimality? In *The 22nd International Conference on Artificial Intelligence and Statistics*, 1611–1619.

Chang, X.; Li, Y.; Oymak, S.; and Thrampoulidis, C. 2020. Provable Benefits of Overparameterization in Model Compression: From Double Descent to Pruning Neural Networks. *arXiv preprint arXiv:2012.08749* .

Chizat, L.; Oyallon, E.; and Bach, F. 2019. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, 2933–2943.

Deng, Z.; Kammoun, A.; and Thrampoulidis, C. 2019. A Model of Double Descent for High-dimensional Binary Linear Classification. *arXiv preprint arXiv:1911.05822* .

Dereziński, M.; Liang, F.; and Mahoney, M. W. 2019. Exact expressions for double descent and implicit regularization via surrogate random design. *arXiv preprint arXiv:1912.04533* .

Du, S. S.; Lee, J. D.; Li, H.; Wang, L.; and Zhai, X. 2018. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804* .

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

Goldt, S.; Reeves, G.; Mézard, M.; Krzakala, F.; and Zdeborová, L. 2020. The Gaussian equivalence of generative models for learning with two-layer neural networks. *arXiv preprint arXiv:2006.14709* .

Gunasekar, S.; Woodworth, B. E.; Bhojanapalli, S.; Neyshabur, B.; and Srebro, N. 2017. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, 6151–6159.

Han, S.; Mao, H.; and Dally, W. J. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* .

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, 1135–1143.

Hassibi, B.; and Stork, D. G. 1993. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in neural information processing systems*, 164–171.

Hassibi, B.; Stork, D. G.; and Wolff, G. 1994. Optimal brain surgeon: Extensions and performance comparisons. In *Advances in neural information processing systems*, 263–270.

Hastie, T.; Montanari, A.; Rosset, S.; and Tibshirani, R. J. 2019. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560* .

Hu, H.; and Lu, Y. M. 2020. Universality laws for high-dimensional learning with random features. *arXiv preprint arXiv:2009.07669* .

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, 8571–8580.

Ji, Z.; and Telgarsky, M. 2018. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300* .

Jin, X.; Yuan, X.; Feng, J.; and Yan, S. 2016. Training skinny deep neural networks with iterative hard thresholding methods. *arXiv preprint arXiv:1607.05423* .

Ju, P.; Lin, X.; and Liu, J. 2020. Overfitting Can Be Harmless for Basis Pursuit, But Only to a Degree. *Advances in Neural Information Processing Systems* 33.

Kini, G.; and Thrampoulidis, C. 2020. Analytic study of double descent in binary classification: The impact of loss. *arXiv preprint arXiv:2001.11572* .

LeCun, Y.; Denker, J. S.; and Solla, S. A. 1990. Optimal brain damage. In *Advances in neural information processing systems*, 598–605.

Lee, N.; Ajanthan, T.; and Torr, P. H. 2018. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340* .

Li, M.; Sattar, Y.; Thrampoulidis, C.; and Oymak, S. 2020. Exploring Weight Importance and Hessian Bias in Model Pruning. *arXiv preprint arXiv:2006.10903* .

Liang, T.; and Rakhlin, A. 2018. Just interpolate: Kernel" ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387* .

Liang, T.; and Sur, P. 2020. A precise high-dimensional asymptotic theory for boosting and min-l1-norm interpolated classifiers. *arXiv preprint arXiv:2002.01586* .

Malach, E.; Yehudai, G.; Shalev-Shwartz, S.; and Shamir, O. 2020. Proving the Lottery Ticket Hypothesis: Pruning is All You Need. *arXiv preprint arXiv:2002.00585* .

Mei, S.; and Montanari, A. 2019. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355* .

Miolane, L.; and Montanari, A. 2018. The distribution of the lasso: Uniform control over sparse balls and adaptive parameter tuning. *arXiv preprint arXiv:1811.01212* .

Montanari, A.; Ruan, F.; Sohn, Y.; and Yan, J. 2019. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544* .

Nakkiran, P.; Kaplun, G.; Bansal, Y.; Yang, T.; Barak, B.; and Sutskever, I. 2019. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292* .

Neyshabur, B.; Tomioka, R.; Salakhutdinov, R.; and Srebro, N. 2017. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071* .

Neyshabur, B.; Tomioka, R.; and Srebro, N. 2014. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614* .

Oymak, S. 2018. Learning Compact Neural Networks with Regularization. *International Conference on Machine Learning* .

Oymak, S.; and Soltanolkotabi, M. 2020. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory* .

Oymak, S.; Thrampoulidis, C.; and Hassibi, B. 2013. The squared-error of generalized lasso: A precise analysis. In *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 1002–1009. IEEE.

Oymak, S.; and Tropp, J. A. 2018. Universality laws for randomized dimension reduction, with applications. *Information and Inference: A Journal of the IMA* 7(3): 337–446.

Pensia, A.; Rajput, S.; Nagle, A.; Vishwakarma, H.; and Papailiopoulos, D. 2020. Optimal Lottery Tickets via SubsetSum: Logarithmic Over-Parameterization is Sufficient. *arXiv preprint arXiv:2006.07990* .

Salehi, F.; Abbasi, E.; and Hassibi, B. 2019. The impact of regularization on high-dimensional logistic regression. In *Advances in Neural Information Processing Systems*, 12005–12015.

Salehi, F.; Abbasi, E.; and Hassibi, B. 2020. The Performance Analysis of Generalized Margin Maximizers on Separable Data. In *International Conference on Machine Learning*, 8417–8426. PMLR.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Stojnic, M. 2013. A framework to characterize performance of LASSO algorithms. *arXiv preprint arXiv:1303.7291* .

Sze, V.; Chen, Y.-H.; Yang, T.-J.; and Emer, J. S. 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE* 105(12): 2295–2329.

Taheri, H.; Pedarsani, R.; and Thrampoulidis, C. 2020. Fundamental limits of ridge-regularized empirical risk minimization in high dimensions. *arXiv preprint arXiv:2006.08917* .

Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; and Le, Q. V. 2019. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2820–2828.

Thrampoulidis, C.; Abbasi, E.; and Hassibi, B. 2018. Precise Error Analysis of Regularized M-Estimators in High Dimensions. *IEEE Transactions on Information Theory* 64(8): 5592–5628.

Thrampoulidis, C.; Oymak, S.; and Hassibi, B. 2015. Regularized linear regression: A precise analysis of the estimation error. In *Conference on Learning Theory*, 1683–1709.

Thrampoulidis, C.; Xu, W.; and Hassibi, B. 2018. Symbol error rate performance of box-relaxation decoders in massive mimo. *IEEE Transactions on Signal Processing* 66(13): 3377–3392.

Tsigler, A.; and Bartlett, P. L. 2020. Benign overfitting in ridge regression. *arXiv preprint arXiv:2009.14286* .

Wang, C.; Zhang, G.; and Grosse, R. 2020. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376* .

Zhou, H.; Lan, J.; Liu, R.; and Yosinski, J. 2019. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *Advances in Neural Information Processing Systems*, 3592–3602.