Ensembling With a Fixed Parameter Budget: When Does It Help and Why?

Didan Deng DDENG@CONNECT.UST.HK

Bertram E. Shi EEBERT@UST.HK

Hong Kong University of Science and Technology, Kowloon, Hong Kong

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Given a fixed parameter budget, one can build a single large neural network or create a memory-split ensemble: a pool of several smaller networks with the same total parameter count as the single network. A memory-split ensemble can outperform its single model counterpart (Lobacheva et al., 2020): a phenomenon known as the memory-split advantage (MSA). The reasons for MSA are still not yet fully understood. In particular, it is difficult in practice to predict when it will exist. This paper sheds light on the reasons underlying MSA using random feature theory. We study the dependence of the MSA on several factors: the parameter budget, the training set size, the L2 regularization and the Stochastic Gradient Descent (SGD) hyper-parameters. Using the bias-variance decomposition, we show that MSA exists when the reduction in variance due to the ensemble (*i.e.*, *ensemble gain*) exceeds the increase in squared bias due to the smaller size of the individual networks (*i.e.*, *shrinkage cost*). Taken together, our theoretical analysis demonstrates that the MSA mainly exists for the small parameter budgets relative to the training set size, and that memory-splitting can be understood as a type of regularization. Adding other forms of regularization, *e.g.* L2 regularization, reduces the MSA. Thus, the potential benefit of memory-splitting lies primarily in the possibility of speed-up via parallel computation. Our empirical experiments with deep neural networks and large image datasets show that MSA is not a general phenomenon, but mainly exists when the number of training iterations is small.

Keywords: Memory-Split Advantage; Deep Ensemble; Regularization.

1. Introduction

The memory-split advantage (MSA), first introduced by Lobacheva et al. (2020) on image recognition tasks, refers to the better performance of an ensemble of several small networks compared to a single large network with the same total parameter count. They call such an ensemble a memory split. A memory-split ensemble is especially beneficial in applications where the parameter budget is limited, *e.g.* by available memory, or on devices capable of parallel computation. The memory-split ensemble usually requires much less time than its single network counterpart.

Concurrently, the memory-split ensemble was investigated empirically in Kondratyuk et al. (2020). The authors found that ensembles could result in higher accuracy than single models and require fewer FLOPs (floating-point operations). They reached the conclusion that the MSA effect consistently exists on large image classification datasets, such as CIFAR-10, CIFAR-100 (Krizhevsky and Hinton, 2009) and ImageNet (Deng et al., 2009). They found the MSA effect also exists for modern CNN architectures, such as ResNet (He et al., 2016) and VGG (Simonyan and Zisserman, 2014). Chirkova et al. (2020) found that besides the common CNN architectures, Transformers (Vaswani et al., 2017) also exhibit MSA for a wide range of parameter budgets on machine translation tasks. Similarly, Dutt et al. (2020) proposed to reallocate the model parameters into several parallel branches at the global network level, and named them as "coupled ensembles". Although phrased differently, "coupled ensembles" are similar to memory-split ensembles. They found that the use of branches improved performance significantly on CIFAR-10, CIFAR-100 and SVHN (Netzer et al., 2011) with ResNet, ResNeXt (Xie et al., 2017), DenseNet-BC (Huang et al., 2017), *etc*.

The memory-split advantage has been witnessed on various dataset-architecture pairs. It seems that the MSA is a generic phenomenon in many modern datasets with common model architectures. However, most past work has been empirical and under limited conditions. The exact reasons and conditions required for the MSA have not been clearly understood yet, which prevents its utilization in various applications. In addition, past work only studied the MSA for classification tasks, not for regression tasks.

In this work, we perform both the theoretical study and the empirical study of the MSA effect mainly on regression tasks. In the theoretical study, we adopt the random feature theory to study the reasons behind the MSA. In the empirical study, we investigate several model architectures on both the toy dataset and large-scale image datasets. Our main contributions are as follows:

- Using bias-variance decomposition of the mean squared loss, we show that MSA exists when the variance reduction from the ensemble (*i.e.*, *ensemble gain*) exceeds the squared bias increase from the shrinking network size (*i.e.*, *shrinkage cost*).
- We theoretically and empirically evaluate the dependence of the MSA on many factors: the parameter budget, the training set size, L2 regularization, the batch size, the learning rate, and number of iterations.
- Our analysis shows the mechanism leading to the MSA is similar to that of other regularization methods, such as L2 regularization.

Definition and notion. In this work, we define the memory-split ensemble as an ensemble consisting of K neural networks: $\{\hat{f}_k\}_{k=1}^K$. Each network has the same architecture and is trained on the same dataset. They all start with random initialization, which leads to different parameters $\{\theta_k\}_{k=1}^K$ after convergence. Given an input x, the output of the memory-split ensemble is $\frac{1}{K}\sum_{k=1}^K \hat{f}_k(x)$ (regression) or $\frac{1}{K}\sum_{k=1}^K \sigma(\hat{f}_k(x))$ (classification, $\sigma(\cdot)$ is an activation function). Broadly speaking, the single model counterpart of the memory-split ensemble should be any single neural network with the same parameter size as the ensemble. However, to be consistent with the related empirical studies, we define the single model counterpart as the single neural network with larger width than the ensemble member $\hat{f}_k$. In other words, the single model counterpart has a similar architecture as $\hat{f}_k$, but with a larger number of channels or hidden units at each layer.

2. Preliminaries

2.1. Random Feature Model

A random feature (RF) model is a simple neural network with only two layers. The first-layer weights are randomly drawn and fixed, while the second layer weights have an explicit solution for a quadratic objective function. RF models are theoretically appealing as they can be viewed as a randomized approximation to kernel ridge regression and are analytically tractable. Therefore, we adopt the random feature theory and apply it to the memory-split ensemble.

Assume the training data $X \in \mathbb{R}^{N \times D}$ are i.i.d. drawn from a Gaussian distribution $\mathcal{N}(0, 1)$, where N is the number of samples and D is the feature dimension. The data-label pair (x, y) is drawn from a linear function corrupted by random noise:

$$y = \langle \beta, x \rangle + \epsilon, \tag{1}$$

where the vector β characterizes the linear relationship, and the additive noise is $\epsilon \sim \mathcal{N}(0, \tau^2)$. We assume the signal to noise ratio is $SNR = F/\tau$. F is the Frobenius norm of β: $F = \|\beta\|$.

We consider a nonlinear random feature model $\hat{f}$ containing two layers. The hidden size is P. An input vector $x \in \mathbb{R}^D$ is fed into the RF model. The output of $\hat{f}$ is

$$\hat{f}(x) = \sum_{i=1}^{P} a_i \sigma\left(\frac{\langle \theta_i, x \rangle}{\sqrt{D}}\right), \tag{2}$$

where $\theta \in \mathbb{R}^{P \times D}$. θ_i is the i^{th} vector in the first-layer weights; σ is the $ReLU$ activation function; and a_i is the i^{th} element of the second-layer weights $a \in \mathbb{R}^P$.

The second-layer weight is solved to minimize the ridge regression loss:

$$\mathcal{L}_{RF} = \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \hat{f}(x_n)\right)^2 + \frac{P\lambda}{D} \|a\|_2^2, \tag{3}$$

where λ is the L2 regularization coefficient.

Note that the solution to Equation (3) is given by

$$a = \frac{1}{\sqrt{D}} y^T Z \left(Z^T Z + \frac{PN}{D^2}\lambda\right)^{-1} \in \mathbb{R}^P, \tag{4}$$

where Z is the first layer output:

$$Z = \frac{1}{\sqrt{D}} \sigma\left(\frac{1}{\sqrt{D}} X \theta^T\right) \in \mathbb{R}^{N \times P}. \tag{5}$$

.

2.2. Bias-Variance Decomposition

Given a training set X, additive noise ϵ, and a random initialization of the weights θ, we can obtain a learned function $\hat{f}$. The squared error between $\hat{f}$ and y on the test set can be decomposed to the sum of the variance term, the bias term and the test noise variance:

$$\mathbb{E}_{x^*}\left[\left(\langle \beta, x^* \rangle + \tilde{\epsilon} - \hat{f}(x^*)\right)^2\right] = \underbrace{\mathbb{E}_{x^*}\left[\left(\langle \beta, x^* \rangle - \mathbb{E}_{X,\theta,\epsilon}\left[\hat{f}(x^*)\right]\right)^2\right]}_{bias^2} + \underbrace{\mathbb{E}_{x^*}\left[\mathbb{E}_{X,\theta,\epsilon}\left[\hat{f}(x^*)^2\right] - \mathbb{E}_{X,\theta,\epsilon}\left[\hat{f}(x^*)\right]^2\right]}_{variance} + \tilde{\tau}^2 \tag{6}$$

Note that x^* is the sample from the test set, where the labels are corrupted by a new noise $\tilde{\epsilon} \sim \mathcal{N}(0, \tilde{\tau}^2)$. The test noise does not affect the decomposition, so it will be set to zero for the rest of our paper.

When $\hat{f}$ is the RF model, the bias-variance decomposition has solutions under asymptotic conditions (*i.e.*, $N, P, D \to \infty$, $\frac{P}{D} = \mathcal{O}(1)$, $\frac{N}{D} = \mathcal{O}(1)$). The solution in d'Ascoli et al. (2020) is given by

$$\mathbb{E}_x \left[\left(\langle \beta, x \rangle + \tilde{\epsilon} - \hat{f}(x) \right)^2 \right] = \xi_{Noise} + \xi_{Init} + \xi_{Samp} + \xi_{Bias}. \tag{7}$$

The variance in Equation 7 consists of three parts: the variance from the noise ξ_{Noise}, the variance from the initialization ξ_{Init}, and the noise from sampling ξ_{Samp}. For explicit expressions, we refer readers to d'Ascoli et al. (2020) for more details. The explicit expressions are closely related to three quantities: $\frac{N}{D}$, $\frac{P}{N}$ and λ. To analyze memory-split ensemble of RF models, we introduce another quantity K, *i.e.*, the number of splits or the ensemble size, to Equation 7. The influence of these quantities to MSA will be extensively discussed in our theoretical analysis.

2.3. Understanding Deep Models with Random Features

Despite the popularity and the capacity of deep learning methods (*i.e.*, deep neural networks), researchers have limited understanding of the mechanism behind their generalization performance. Recently, a spate of papers (Du et al., 2018; Cao and Gu, 2019; Allen-Zhu et al., 2019; Du et al., 2019) observed a phenomenon named "lazy-training": for over-parameterized neural networks, standard gradient-based methods change the model weights very slowly, making them very close to their initial random values (almost fixed). Fixing certain weights resembles the learning with random features (Yehudai and Shamir, 2019). Chizat et al. (2018) proved that the lazy training phenomenon leads to a model equivalent to learning with positive-definite kernels. In fact, random feature models were initially proposed as a computationally-efficient alternative to kernel methods (Rahimi et al., 2007). It has been mathematically proved that a Gaussian RF model is close to a kernel predictor (Jacot et al., 2020).

Recent work (Spigler et al., 2019; Yang et al., 2020) has established links between RF models and deep neural networks. For example, RF models explain the reasons for double-descent curves, which also appear in deep models (Mei and Montanari, 2019). Our work builds upon this by pointing out additional connections, while also recognizing the gap between theory and empirical observations. Firstly, the features of our synthesized dataset for RF models are Gaussian i.i.d.. These features usually do not exist in real datasets. Secondly, the closed-form expressions for the generalization error with RF models hold only asymptotically. This condition can not be satisfied in practice. Despite these limitations, we seek to explore the similarities and dissimilarities between theory and empirical results, and provide valuable insights on MSA.

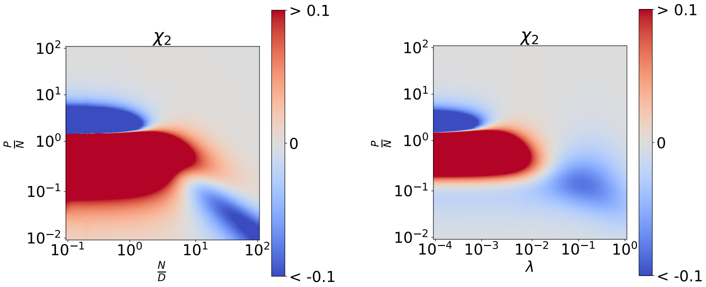
3. Theoretical Analysis of MSA

A memory-split ensemble is denoted by $\hat{F}_K = \frac{1}{K} \sum_{k=1}^{K} \hat{f}_k$ (regression). K is the ensemble size, and $\hat{f}_k$ is the k^{th} model trained with ridge regression. In our theoretical analysis, $\hat{f}_k$ is the RF model. Because the parameter size of the RF model is linearly related to the hidden size, we assume the single model counterpart to have P hidden units, and each RF model in a K-split ensemble to have $\frac{P}{K}$ hidden units.

The MSA exists when the memory-split ensemble ($K > 1$) has a lower generalization error than its single model counterpart (*i.e.*, $K = 1$). For simplicity, we use the generalization error difference between $\hat{F}_1$ and $\hat{F}_K$ as an indicator of the MSA's existence:

$$\chi_K = \mathbb{E}_x \left[\left(y - \hat{F}_1(x) \right)^2 \right] - \mathbb{E}_x \left[\left(y - \hat{F}_K(x) \right)^2 \right]. \tag{8}$$

$\hat{F}_1(x)$ and $\hat{F}_K(x)$ have the same number of parameters. If χ_K is positive, the MSA exists for the split K. If χ_K is negative, the MSA does not exist for the split K. For ease of expression, we use "the existence of the MSA" and "the existence of the MSA for split $K = 2$" interchangeably if not specifying K.



(*a*) $\lambda = 10^{-2}$ (*b*) $\frac{N}{D} = 10$

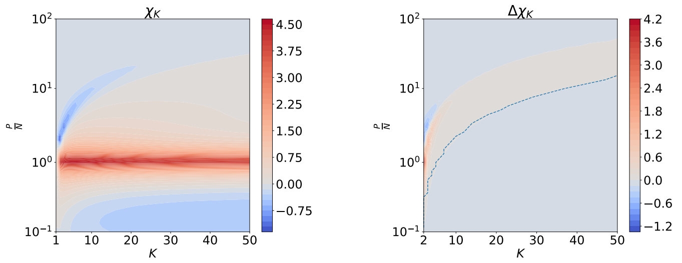
Figure 1: The MSA's existence: (a) the influence of $\frac{N}{D}$ and $\frac{P}{N}$ on the MSA under the condition that $\lambda = 10^{-2}$; (b) the influence of λ and $\frac{P}{N}$ on the MSA under the condition that $\frac{N}{D} = 10$. Increasing $\frac{N}{D}$ and increasing λ have similar effects on the MSA.

3.1. The Interplay

In our analysis, we find that the MSA stems from the interplay between four factors: the ratio of the training set size to the feature dimension $\frac{N}{D}$, the ratio of the parameter size to the training set size $\frac{P}{N}$, the L2 regularization λ and the split size K. For a single RF model, P is its hidden size, while for a memory-split ensemble model, P is the total number of trainable parameters. In our analysis, we will fix two of the four factors and plot the influence of the remaining two factors on the MSA. To obtain the results, we set $F = 1, SNR = F/\tau = 1$. Without specification, we used the same F and SNR for the rest of this paper.

3.1.1. THE EXISTENCE OF MSA

In Figure 1, we plot the value of χ_2 under two different conditions: $\lambda = 10^{-2}$ and $\frac{N}{D} = 10$. The red color indicates the existence of the MSA effect, while the blue color indicates the memory-split ensemble performs worse. A great portion of images is covered by the neutral color, which indicates the memory-split ensemble has the same test error as its single model counterpart.



(*a*) $\chi_K, K \in \{1, 2, ..., 50\}$ (*b*) $\Delta\chi_K, K \in \{1, 2, ..., 50\}$

Figure 2: The χ_K and $\Delta\chi_K$ for $K \in \{1, 2, ..., 50\}$. (a): the influence of $\frac{P}{N}$ and K on the MSA. (b): the influence of $\frac{P}{N}$ and K on the MSA increment. The color-map is centered at zero.

From Figure 1 (a), we find MSA exists mainly for small parameter size compared to the training set size ($\frac{P}{N} \leq 1$) and small training set size compared to the feature dimension ($\frac{N}{D} \leq 10$). Similarly, from Figure 1 (b), we find that increasing $\frac{P}{N}$ and increasing λ both eliminate MSA. Based on Figure 1, we conclude that the MSA effect is not a generic phenomenon for RF models. Large regularization, large parameter size, and large training set size prevent the existence of MSA. Surprisingly, our findings are inconsistent with the empirical findings for deep neural networks in Lobacheva et al. (2020); Kondratyuk et al. (2020); Chirkova et al. (2020). They witnessed the deep neural networks exhibited the MSA effect even for the large training set and large parameter size. This suggests that the MSA for deep models may be influenced by other factors besides the aforementioned four factors, which will be discussed later.

We demonstrate the MSA and the MSA increment for larger splits in Figure 2 when $\frac{N}{D} = 10, \lambda = 10^{-4}$. $\Delta\chi_K$ is the generalization performance increment from the split $K-1$ to the split K, which is defined as:

$$\Delta\chi_K = \chi_K - \chi_{K-1}. \tag{9}$$

From Figure 2 (a), we find that the MSA existence for $K = 2$ does not indicate the MSA existence for large splits ($K > 2$), but MSA always exists at $\frac{P}{N} = 1$ for $K \in \{2, ..., 50\}$. This parameter size $\frac{P}{N} = 1$ is named as the "transition point" by Spigler et al. (2019). It has been proved that there exists a variance peak at the transition point, which coincides with the MSA peak at $\frac{P}{N}$ (Spigler et al., 2019).

In Figure 2 (b), the red color indicates that the test performance is improved from $K-1$ to K. We highlight the optimal K where the largest memory-split advantage is obtained for different $\frac{P}{N}$ using the dashed curve. Except for some extremely small values of $\frac{P}{N}$, the optimal K grows with the increase of parameter size. We find that the optimal K grows approximately linearly with $\frac{P}{N}$. This finding is consistent with the Figure 6 in Lobacheva et al. (2020), which shows that the optimal split exhibits a nearly linear relationship with the total parameter counts for VGG and WideResNet (Zagoruyko and Komodakis, 2016) on the CIFAR-100 dataset.

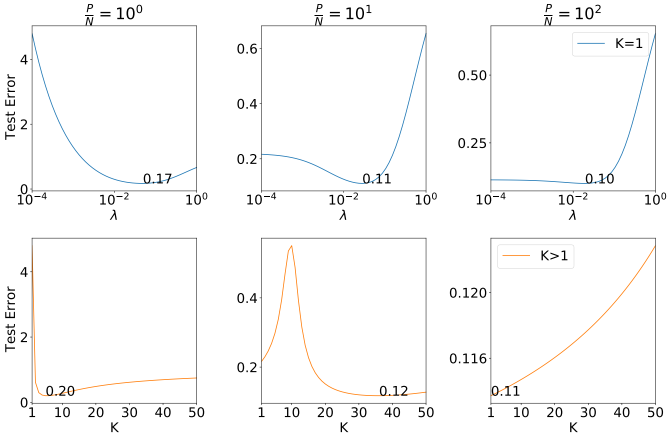


Figure 3: Comparison between L2 regularization (top row) and the memory-split ensemble (bottom row) for $\frac{P}{N} \in \{1, 10, 10^2\}$, $\frac{N}{D} = 10$. The test error at the optimal K is slightly worse than the test error at the optimal λ.

3.1.2. THE L2 REGULARIZATION AND MEMORY-SPLIT ENSEMBLE

We compare the test errors using L2 regularization or memory-split ensemble. In the first row of Figure 3, we use a single model ($K = 1$) and vary the L2 regularization coefficient λ from 10^{-4} to 1. The text in each plot shows the lowest test error over λ. We do not consider $\frac{P}{N} < 1$ since over-parameterization is more common in modern deep learning than under-parameterization.

In the second row of Figure 3, we use a memory-split ensemble ($K > 1$) and fix the L2 regularization $\lambda = 10^{-4}$. The text in each plot shows the lowest test error over K. The memory-split ensemble leads to similar results as L2 regularization, but it does not lead to better generalization performance. For the largest parameter size $\frac{P}{N} = 10^2$, the memory-split ensemble leads to worse generalization performance as K increases. The memory-split ensemble can only be varied discretely. The strongest advantage of the memory-split ensemble over L2 regularization is that it can be deployed in parallel computation systems naturally.

3.2. The Reason for the MSA

Although the aforementioned four factors are related to the MSA, they are not the direct reasons. All four factors can alter both the bias and variance, which leads to the change of MSA.

Figure 1 (a) plots the values of χ_2 given various pairs of $\left(\frac{N}{D}, \frac{P}{N}\right)$ for $\lambda = 10^{-2}, K = 2$. We identity three typical phases of the MSA from this plot. Under the condition that $\frac{N}{D} = 10^{-1}$, increasing $\frac{P}{N}$ leads to the shift from positive χ_2 to negative χ_2. Under the condition that $\frac{N}{D} = 10^{0.5}$, χ_2 is positive for all the $\frac{P}{N}$ we consider. Under the condition that $\frac{N}{D} = 10$, we witness a shift from negative χ_2 to positive χ_2 as we increase $\frac{P}{N}$. We name the three phases as Phase I, II and III respectively. The three phases of MSA characterize distinct generalization behaviors due to the change of training set size (or $\frac{N}{D}$). We can identity similar phases due to the change of λ in Figure 1 (b).

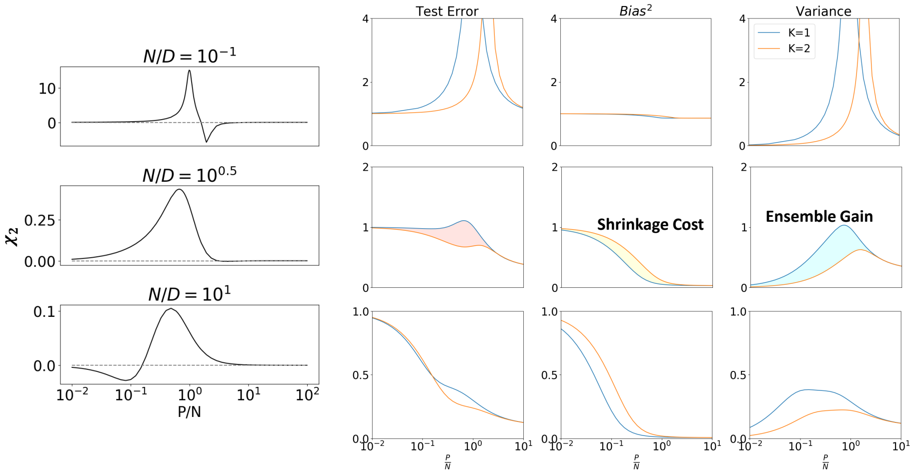


Figure 4: The three phases of the MSA: Phase I, II and III from top to bottom. The left-hand figures plot the values of χ_2 for different $\frac{N}{D}$ and $\frac{P}{N}$ ($\lambda = 10^{-2}$), where positive values indicate the existence of the MSA. The right-hand images shows the bias-variance decomposition for the three phases of the MSA. For Phase II, the *Shrinkage Cost* and *Ensemble Gain* are highlighted using shading.

In Figure 4, we plots the χ_2 values in three phases on the left-hand side and their bias-variance decomposition on the right-hand side. From Phase I to Phase III, the peak value of χ_2 gets smaller. This indicates that the memory-split ensemble results in less performance gain when the training set size is large. Comparing the bias and variance for $K = 1$ and $K = 2$, we notice two major differences. Firstly, the ensemble bias is greater than the single model bias. We refer to the increase in bias as the *shrinkage cost*. Secondly, the ensemble has smaller variance than the single model. We refer to the reduction of the variance as the *ensemble gain*. When the *ensemble gain* exceeds the *shrinkage cost*, the MSA exists. Based on the analysis above, we draw the conclusion that the reason for the MSA is that the *ensemble gain* exceeds the *shrinkage cost*. Predicting the existence of the MSA is difficult because the complex interplay between many factors leads to different phases of the MSA.

4. Numerical Experiments

4.1. Estimating Bias and Variance

For a clear comparison between theoretical results and numerical results, we modified a method in Yang et al. (2020) to empirically estimate the bias and variance of any memory-split ensemble. The pseudo code is given by Algorithm 1.

In Algorithm 1, we run a large number of trials ($T = 50$) to estimate the bias square term $Bias^2$ and the variance term Var. This algorithm can be applied to any model, *e.g.*, random feature models and neural networks. P denotes the total number of trainable parameters in the ensemble. This algorithm can be applied to the synthetic dataset defined as Equation (1) and to real datasets.

Algorithm 1 Estimating Bias and Variance

```
input  : Training set $\mathcal{S}$ and test set $x$.
         The number of trials $T$.
         The ensemble size $K$.
         The training set size $N$.
         The feature dimension $D$.
         The parameter size $P$
output: The bias term $Bias^2$ and the variance term $Var$.

for $t \leftarrow 1$ to $T$ do
    Randomly sub-sample $N$ data points from $\mathcal{S}$ as training data $X$;
    Randomly sample additive noise $\epsilon$ to corrupt the training labels;
    for $k \leftarrow 1$ to $K$ do
        Randomly initialize $\theta_k$ for the $k^{th}$ model $\hat{f}_k$;
        Optimize $\hat{f}_k$;
    end
    $\hat{F}_K = \frac{1}{K}\sum_{k=1}^{K} \hat{f}_k$;
end
$Bias^2 = \mathbb{E}_x\left[\left(\langle\beta, x\rangle - \mathbb{E}_{X,\theta,\epsilon}\left[\hat{F}_K(x)\right]\right)^2\right]$; $Var = \mathbb{E}_x\left[\mathbb{E}_{X,\theta,\epsilon}\left[\hat{F}_K(x)^2\right] - \mathbb{E}_{X,\theta,\epsilon}\left[\hat{F}_K(x)\right]^2\right]$.
```

When sampling real datasets, we assume the noise is already embedded in X, and therefore we do not add additional noise ϵ.

4.2. Random Feature Model Experiments

In this section, we run experiments with random feature models using Algorithm 1. We show the extent to which the theoretical results agree with the empirical results on both the synthetic dataset and the real dataset.

4.2.1. On the synthetic dataset

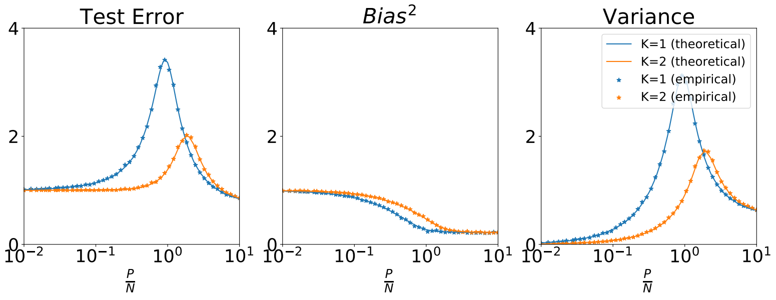


Figure 5: The generalization error, $Bias^2$ and variance of the random feature model theoretically (solid lines) and empirically (stars). $\frac{N}{D} = 1, \lambda = 10^{-2}$.

To synthesize training data, we use $D = 400, \frac{N}{D} = 1, \lambda = 10^{-2}$. We plot both the theoretical results and the empirical results in Figure 5. This shows that the theoretical results agree with the empirical results even at a moderate feature dimension ($D = 400$) despite the fact that the theoretical results are under asymptotic conditions. In addition, the results show that the Algorithm 1 can estimate the bias and variance very accurately.

4.2.2. ON THE REAL DATASET

We choose to run RF models on the MNIST dataset (LeCun and Cortes, 2010). The input feature dimension D is 784 since we flatten the input image into a feature vector. The training set is a subset of the MNIST dataset ($N = 784$). The test set is the official test set consisting of 10000 images.

Learning with ridge regression. We train the random feature models using ridge regression (Equation (4)). The empirical results for $\frac{N}{D} = 1, \lambda = 10^{-2}$ are shown in Figure 6.

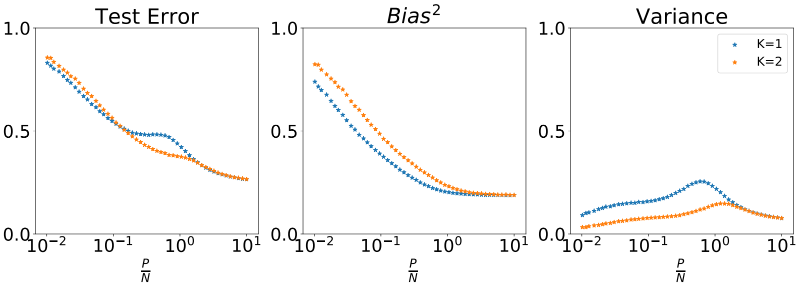


Figure 6: The empirical results of the random feature model trained on MNIST dataset using ridge regression. $\frac{N}{D} = 1, \lambda = 10^{-2}$.

From Figure 6, we find that the empirical results are close to the Phase III in Figure 4. Note that the theoretical results on the synthetic dataset for $\frac{N}{D} = 1, \lambda = 10^{-2}$ should be in between the Phase I and the Phase II (Figure 5). We believe the reason for the mismatch is that, for the MNIST dataset, adjacent pixels are closely related. Therefore, the intrinsic dimension (*i.e.*, the number of variables needed in a minimal representation of the data) is much smaller than 784. The effective $\frac{N}{D}$ of the MNIST dataset should be greater than 1, which is the reason why Figure 6 is close to Phase III.

Learning with SGD. Random feature theory does not consider noise induced by Stochastic Gradient Descent (SGD). Recent studies confirm that the noise induced by SGD are closely related to the batch size, the learning rate. Jastrzębski et al. (2017) show that the ratio of the learning rate to batch size, which changes the SGD noise magnitude, has a great influence on the generalization of SGD: the larger the ratio, the better the generalization. Hoffer et al. (2017) also claim that large-batch training can generalize as well as small-batch training by increasing the number of iterations. This suggests that increasing the number of iterations also affects generalization.

To investigate the effect of SGD hyper-parameters to the MSA, we train the random feature models with mini-batch SGD. We vary the ratio of the learning rate to batch size; and change the number of iterations. During training, we fix the first-layer weight and only train the second-layer weight using SGD. The feature dimension is 784. For $\frac{N}{D} = 1, \lambda = 10^{-2}$, we choose batch size from $b \in \{10, 784\}$ and the number of iterations from $\Gamma \in \{500, 5000\}$. The optimizer that we use

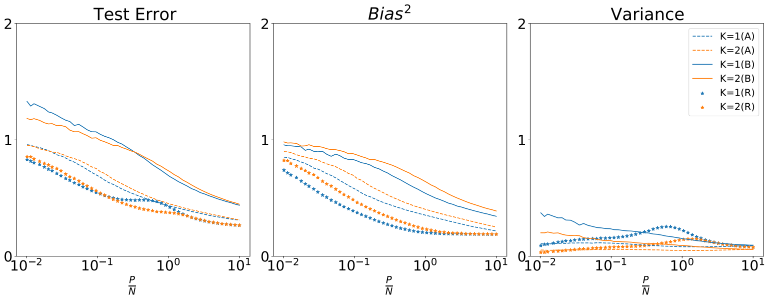


Figure 7: The empirical results of the random feature model trained on MNIST dataset using mini-batch SGD (A and B) and ridge regression (R). $\frac{N}{D} = 1, \lambda = 10^{-2}$.

is the SGD optimizer (momentum $= 0.9$). The learning rate is fixed as 0.01. There are two settings of SGD that we compare: (A) $b = 10, \Gamma = 5000$, and (B) $b = 784, \Gamma = 500$.

Figure 7 shows the results for settings (A) and (B) using SGD, as well as the results obtained using ridge regression, which is denoted by (R). For setting (A), we find no MSA for all the parameter budgets we consider. Training over a large number of iterations suppresses variance significantly, so the ensemble does not benefit too much from reducing variance. For setting (B), the models are not trained sufficiently, and thus both the variance and bias are high. For some parameter budgets in setting (B), we find the existence of the MSA. These results prove that besides the aforementioned factors, the hyper-parameters of SGD also affect the MSA. This makes the MSA's existence difficult to predict in real applications since it is influenced by many factors.

4.3. Neural Network Experiments

In the experiments with deep neural networks and real datasets, we investigate the existence of MSA for multiple parameter budgets and compare the results under small and large number of iterations.

There are two model architectures we consider: ResNet18 (He et al., 2016) and the Vision Transformer (ViT) (Dosovitskiy et al., 2020). ViT replaces convolutional filters with self-attention modules (Vaswani et al., 2017) and has been proved to have a comparable performance with CNN.

To change the parameter size, we vary the width factor w. For ResNet18 model, w is the number of filters in its first convolutional block. ResNet18 has $[w, 2w, 4w, 8w]$ filters sequentially. When the width factor of ResNet18 equals to 64, the parameter size P is 11.5 million and we refer to it as the standard parameter budget P_s. We consider $8 \leq w \leq 64$, which corresponds to $\frac{1}{64}P_s \leq P \leq P_s$. For ViT, we define the architecture by setting the number of layers and the number of heads in the Transformer encoder to 6 and 8 respectively. The width parameter w is the feature size of patch embedding and the hidden size of the second last fully connected layer. When $w = 512$, ViT model has about 9.8 million parameters, and we refer to this model size as the standard parameter budget P_s for ViT. We consider $64 \leq w \leq 512$, which corresponds to $\frac{1}{64}P_s \leq P \leq P_s$ for ViT.

For each width factor w, we train at least $l = 16$ networks, and construct $\lfloor \frac{l}{2} \rfloor$ ensembles. Each ensemble consists of two distinct networks. To compare the test error of the single model ($K = 1$) and the two-split ensemble ($K = 2$), we average the mean squared error over $\lfloor \frac{l}{2} \rfloor$ runs. We train all the networks with the Adam optimizer without L2 regularization. The initial learning rate is 10^{-3}

for ResNet18 and 10^{-4} for ViT. The cosine annealing schedule (Loshchilov and Hutter, 2016) is used to improve convergence.

We used two datasets: the CIFAR-10 (Krizhevsky and Hinton, 2009) dataset and the MPII Gaze dataset (Zhang et al., 2017). The CIFAR-10 dataset has 50000 training samples and 10000 test samples. The image size is 32×32. The batch size is 128. The MSA existence is shown by comparing the validation accuracy for $K = 1$ and $K = 2$ under different parameter budgets P and different number of training iterations N_{iter}. We denote the number of training epochs as N_{epoch}.

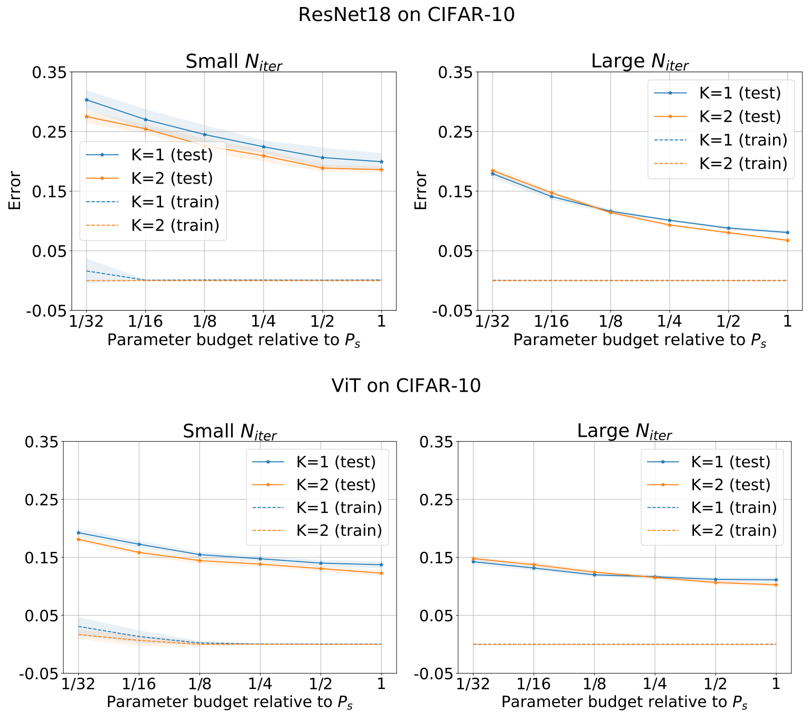


Figure 8: The experiment results with ResNet18 and ViT on CIFAR-10 dataset. For ResNet18, the small N_{iter} corresponds to $N_{epoch} = 200$, and the large N_{iter} corresponds to $N_{epoch} = 1000$. ViT takes longer to converge (*i.e.*, the training error is close to zero). Therefore, the small N_{iter} corresponds to $N_{epoch} = 1000$, and the large N_{iter} corresponds to $N_{epoch} = 5000$ for ViT. We use shadows to show the standard deviations of the results.

The experiment results on the CIFAR-10 dataset are shown in Figure 8. We plot the classification error for single models ($K = 1$) in blue and memory-split ensembles ($K = 2$) in orange. The test errors are shown as solid curves, and the training errors are shown as dashed curves. For both ResNet18 and ViT, we find that MSA exists when the number of training iterations is small. Under insufficient training, we think the model weights scatter around the global minimum, which leads to variance in the test errors. The memory-split ensemble can reduce this variance. Therefore, we find that the memory-split ensemble generalizes better than the single model for all the parameter budgets we consider when N_{iter} is small. The results for small N_{iter} in Figure 8 are consistent with the findings in Lobacheva et al. (2020); Kondratyuk et al. (2020) and Chirkova et al. (2020).

However, we have additional findings about MSA's existence for large N_{iter}, which have not been explored by previous studies.

When N_{iter} is large, we train the models for extra iterations after it converges (*i.e.*, the training errors close to zero). This may push the model weights closer to the global minimum, so the model weights have less variance. We notice that the MSA is less obvious for large N_{iter} than for small N_{iter}. The MSA does not exist for small parameter budgets (*e.g.*, $\frac{1}{32}P_s$), but it exists for large parameter budgets (*e.g.*, P_s). The results of large N_{iter} resemble Phase III in Figure 4 obtained from the theoretical analysis of RF models.

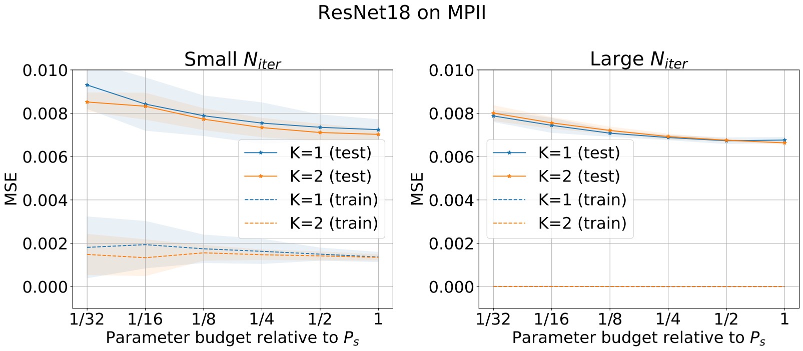


Figure 9: The experiment results with ResNet18 on MPIIGaze dataset. We use shadows to show the standard deviations of the results.

The MPII Gaze dataset is a dataset for appearance-based gaze estimation in the wild. It is separated into 15 parts by the identities of subjects. Each subject has 3000 images. We take the first 12 subjects' images as the training set and use the remaining images as the test set. No validation set or early stopping is used. The batch size is 128. We choose N_{epoch} from $\{200, 1000\}$, which corresponds to the small and large N_{iter}. The input image is a $(256, 64)$ RGB image. Each input image contains two eyes.

The experiment results with ResNet18 on the MPII gaze dataset are shown in Figure 9. Similar to Figure 8, the MSA exists for all considered parameter budgets when N_{iter} is small. When N_{iter} is large, the MSA is less obvious and does not exist for small parameter budgets. The existence of MSA has similar patterns for classification tasks and regression tasks with deep neural networks and real datasets.

5. Related Works

Memory-split Ensemble. The memory-split ensembles, although have been empirically studied, have not been theoretically analyzed in related papers. Kondratyuk et al. (2020), Dutt et al. (2020), Lobacheva et al. (2020) and Chirkova et al. (2020) provided empirical evidence to the existence of MSA for deep models and large datasets. However, they lacked a systematic experimental setting to verify the generality of the MSA for both simple networks and complex networks. In addition, they failed to investigate this phenomenon theoretically. Webb et al. (2020) found empirically that a single ”big” model outperforms an ensemble for various parameter budgets, where they considered the influence of parameter budgets (P in our work). Our work presents a more extensive study of related factors. Furthermore, we tried to explain the emergence of MSA using RF theory.

Bias-variance Decomposition. For bias-variance decomposition, Ueda and Nakano (1996) proved that ensemble estimators can benefit from reduced variance. The reduced variance is proportional to the ensemble size K. Although they did not consider a fixed parameter budget, they inspired us to analyze memory-split ensembles using bias-variance decomposition. Our work is closely related to d'Ascoli et al. (2020) because the RF theory is largely based on their explicit expressions for bias and variance terms. The novelty of our work compared to d'Ascoli et al. (2020) is that we extend their RF theory to memory-split ensembles where $K > 1$.

6. Conclusion

In this work, we shed light on the mystery of the memory-split advantage (MSA) using random feature theory and empirical study. Based on our analysis, we find that the MSA stems from the complex interplay between training set size, parameter size, L2 regularization and SGD hyperparameters. Because of this complex interplay, the MSA does not always exist, and it is very hard to predict when it will exist in real applications. The memory-split ensemble works as a regularization method, but we find it to be no better than L2 regularization in terms of the optimal generalization performance. Our source code is available at https://github.com/wtomin/MemorySplitAdvantage.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

Yuan Cao and Quanquan Gu. A generalization theory of gradient descent for learning overparameterized deep relu networks. *arXiv preprint arXiv:1902.01384*, 2, 2019.

Nadezhda Chirkova, Ekaterina Lobacheva, and Dmitry Vetrov. Deep ensembles on a fixed memory budget: One wide network or several thinner ones? *arXiv preprint arXiv:2005.07292*, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

Stéphane d'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent: Bias and variance (s) in the lazy regime. In *International Conference on Machine Learning*, pages 2280–2290. PMLR, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Anuvabh Dutt, Denis Pellerin, and Georges Quénot. Coupled ensembles of neural networks. *Neurocomputing*, 396:346–357, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. *arXiv preprint arXiv:1705.08741*, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Implicit regularization of random feature models. In *International Conference on Machine Learning*, pages 4631–4640. PMLR, 2020.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Dan Kondratyuk, Mingxing Tan, Matthew Brown, and Boqing Gong. When ensembling smaller models is more efficient than single large models. *arXiv preprint arXiv:2005.00570*, 2020.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Ekaterina Lobacheva, Nadezhda Chirkova, Maxim Kodryan, and Dmitry Vetrov. On power laws in deep ensembles. *arXiv preprint arXiv:2007.08483*, 2020.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

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

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Ali Rahimi, Benjamin Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5. Citeseer, 2007.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Stefano Spigler, Mario Geiger, Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Matthieu Wyart. A jamming transition from under-to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, 2019.

Naonori Ueda and Ryohei Nakano. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pages 90–95. IEEE, 1996.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Andrew Webb, Charles Reynolds, Wenlin Chen, Henry Reeve, Dan Iliescu, Mikel Lujan, and Gavin Brown. To ensemble or not ensemble: When does end-to-end training fail? In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 109–123. Springer, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning*, pages 10767–10777. PMLR, 2020.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. *Advances in Neural Information Processing Systems*, 32:6598–6608, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):162–175, 2017.