Journal of Machine Learning Research 23 (2022) 1-40 Submitted 12/21; Revised 11/22; Published 12/22

Impact of classification difficulty on the weight matrices spectra in Deep Learning and application to early-stopping

Xuran Meng U3007800@CONNECT.HKU.HK
Department of Statistics and Acturial Science
The University of Hong Kong

Jianfeng Yao JEFFYAO@CUHK.EDU.CN
School of Data Science
The Chinease University of Hong Kong (Shenzhen)

Editor: Michael Mahoney

Abstract

Much recent research effort has been devoted to explain the success of deep learning. Random Matrix Theory (RMT) provides an emerging way to this end by analyzing the spectra of large random matrices involved in a trained deep neural network (DNN) such as weight matrices or Hessian matrices in the stochastic gradient descent algorithm. To better understand spectra of weight matrices, we conduct extensive experiments on weight matrices under different settings for layers, networks and data sets. Based on the previous work of Martin and Mahoney (2021b), spectra of weight matrices at the terminal stage of training are classified into three main types: Light Tail (LT), Bulk Transition period (BT) and Heavy Tail (HT). These different types, especially HT, implicitly indicate some regularization in the DNNs. In this paper, inspired from Martin and Mahoney (2021b), we identify the difficulty of the classification problem as an important factor for the appearance of HT in weight matrices spectra. Higher the classification difficulty, higher the chance for HT to appear. Moreover, the classification difficulty can be affected either by the signal-to-noise ratio of the dataset, or by the complexity of the classification problem (complex features, large number of classes) as well. Leveraging on this finding, we further propose a spectral criterion to detect the appearance of HT and use it to early stop the training process without testing data. Such early stopped DNNs have the merit of avoiding overfitting and unnecessary extra training while preserving a much comparable generalization ability. These findings from the paper are validated in several NNs (LeNet, MiniAlexNet and VGG), using Gaussian synthetic data and real data sets (MNIST and CIFAR10).

Keywords: Deep Learning, Weight matrices, Heavy tailed spectrum, Early stopping

1. Introduction

In the past decade, deep learning (LeCun et al., 2015) has achieved impressive success in numerous areas. Much research effort has since been concentrated on providing a rational explanation of the success. The task is difficult, particularly because the training of most successful deep neural networks (DNNs) relies on a collection of expert choices that determine the final structure of the DNNs. These expert choices include nonlinear activation, hidden layer architecture, loss function, back propagation algorithm and canonical datasets. Unfortunately, these empirical choices usually bring non-linearity into the model, and non-convexity of optimization into the training process. As a matter of consequence, practitioners of deep learning are facing certain lack of general guide-

lines about the "right choices" to design and train an effective DNN for their own machine learning problem.

To make progress on the understanding of existing trained and successful DNNs, it is important to explore their properties in some principled way. To this end, a popular way has recently emerged in the literature, namely spectral analysis of various large characteristic random matrices of the DNNs, such as the Hessian matrices of the back-propagation algorithm, weight matrices between different layers, and covariance matrices of output features. Actually, such spectral analysis helps to gain insights into the behavior of DNNs, and many researchers believe that these spectral properties, once better understood, will provide clues to improvements in deep learning training (Dauphin et al., 2014; Papyan, 2019b,a; Sagun et al., 2017; Yao et al., 2020; Granziol, 2020; Pennington and Worah, 2019; Ge et al., 2021). Recently, Martin and Mahoney (2021b) studied the empirical spectra distributions (ESD) of weight matrices in different neural networks, and observed a "Phase Transition 5+1" phenomenon in these ESDs. Interestingly, the phenomenon highlights signatures of traditionally regularized statistical models even though there is no set-up of any traditional regularization in the DNNs. Here, traditional regularization refers to the minimization of an explicitly defined and penalized loss function of the form $L(\theta) + \alpha \cdot p(\theta)$ with some tuning parameter α (θ denotes all the parameters in the DNN). However, those well-known expert choices such as early stopping also produces a regularization effect in DNNs, and this is the reason why such expert choices are recommended for practitioners. Actually, Kukacka et al. (2017) presented about 50 different regularization techniques which may improve DNN's generalization. Among them, batch normalization, early stopping, dropout, and weight decay are a few commonly used ones.

A main finding from Martin and Mahoney (2021b) is that the effects of these regularization practices can be identified through the spectra of different weight matrices of a DNN. Moreover, the forms of these spectra in the "5+1 phase transition" help assess certain degree of regularization in the DNN. For instance, if these spectra are far away from the Marčenko-Pastur (MP) law, or the largest eigenvalue departs from the Tracy-Widom (TW) Law (see Appendix A), there is strong evidence for the onset of more regular structures in the weight matrices. A connection between implicit regularization in a DNN and the forms of the spectra of its weight matrices is thus established. Particularly, they considered the evolution of weight matrices spectra during the training process of a DNN from its start to its final stage (usually 200-400 epochs), and pointed out that in late stage of the training, the deviation of the spectra from the MP Law (namely the emergence of Heavy Tail) indicates certain regularization of the DNN, synonym of an improved genelization ability. Indeed, such regularization implies high-correlated entries in the weight matrices and thus leads to a heavy tailed spectrum. Recently, Gurbuzbalaban et al. (2021) pointed out that for linear regressions the SGD can also produce heavy tails in weight matrix spectra. Hodgkinson and Mahoney (2021) on the other hand explored the impact of other factors on the emergence of heavy tails which relate to the optimization process such as increasing the step size/decreasing the batch size, or increasing L_2 regularization.

In Martin et al. (2021), the authors found that the "Heavy Tail based metrics can do much better—quantitatively better at discriminating among series of well-trained models with a given architecture; and qualitatively better at discriminating well-trained versus poorly trained models." Experiments conducted in this research confirm the importance of such heavy tail phenomenon for the understanding of deep learning.

Specifically, we identify a precise factor, that we term as *classification difficulty*, which strongly controls the appearance or not of heavy tails in weight matrix spectra at the final stage of the training.

The greater classification difficulty, the higher possibility that heavy tails appear. Moreover, we showcase two situations of difficult classification that lead to heavy tails. In one situation, the data quality is poor (or its signal-to-noise ration is low) and the emergence of heavy tails indicates an attempt for DNNs to extract more features and increase testing accuracy. The other situation is more related to a higher complexity of the classification problems such as in modern data sets with a large number of features and classes, and the emergence of heavy tails here indicates an attempt for DNNs to identify relevant data features. While both situations have a high classification difficulty and lead to heavy tails in weight matrix spectra, the training results could be entirely different. In the second situation, the emergence of HT indicates a continuous and healthy feature extraction process that gradually improves the test accuracy of the DNN. However, in the first situation, the emergence of heavy tails indicates some excessive information extraction and thus leads to overfitting.

Note that as a factor controlling the heavy tail phenomenon, the classification difficulty differs from the other factors identified in the SGD or the hyper-parameters involved in the optimization process as discussed in Gurbuzbalaban et al. (2021) and Hodgkinson and Mahoney (2021). Intuitively, the classification difficulty is a statistical metric for how difficult classes in a data set can be identified under certain model architectures. Nevertheless the classification difficulty is still a vague concept and may depend on many properties of the data set and model architectures. In this paper, we focus our discussion on two factors that directly impact on the classification difficulty, namely the data quality and the complexity of the classification problem.

As an important application of our observations on the spectrum types and on the emergence of heavy tails, we propose a spectral criterion to guide the early stopping in practice. Without prior information, heavy tails indicate some regularization at play or some problematic issues such as overfitting in the training process. Roughly speaking, we early stop the training when there is statistically significant evidence that heavy tails appear in weight matrix spectra. Such early stopped DNNs have the merit of avoiding overfitting and unnecessary extra training while preserving a much comparable generalization ability. These findings from the paper are validated in several NNs (LeNet, MiniAlexNet and VGG), using Gaussian synthetic data and real data sets (MNIST and CIFAR10). Note that the idea of using evolution of weight matrices to monitor the training process of a DNN has appeared earlier in the AI community with the online WeightWatcher package[1]. However to our best knowledge, our spectral criterion is the first quantitative criterion based on the weight matrix spectra to guide early stopping of a training process.

We summarize our contributions as follows:

1. The difficulty of a classification problem is identified as a driving factor for the appearance of heavy tails in weight matrices spectra. Experiments conducted on both synthetic and real data sets support this finding. Particularly, decreasing the SNR of the data set or increasing the number of classes K in Gaussian data experiments all increase the classification difficulty and generate heavy tails at the end of training. In real data experiments, heavy tails appear more in experiments with CIFAR10 than with MNIST due to more complex features and a higher classification difficulty in CIFAR10.

2. We reformulate the "5+1" classification of Martin and Mahoney (2021b) into a smaller classification of the bulks of weight matrix spectra at final training stage: Light Tail (LT), Bulk Transition period (BT) and Heavy Tail (HT). With a decreasing classification difficulty, these

1. [https://github.com/CalculatedContent/WeightWatcher], a companion package to Martin and Mahoney (2021b).

spectrum bulks obey a phase transition from HT to BT, and then to LT. This simpler classification of spectra types help demonstrate the phase transition phenomenon from HT to BT, and then to LT, a phenomenon widely observed previously and also in our experiments. Our finding of the classification difficulty as the main driving factor of this phase transition is also based on this simpler classification.

3. Leveraging on these findings, we propose a spectral criterion to guide the early stopping without access of testing data. The HT(BT)-based spectral criterion could not only cut off a large training time with just a little drop of test accuracy, but also avoid over-fitting even when the training accuracy is increasing.

The remaining of the paper is organized as follows. Sections 2 and 3 report our experimental results on synthetic data and real data sets, respectively. The spectral criterion for early stopping is introduced in Section 4. Related theoretical developments are put in Appendices A and B of the supplementary materials, and additional algorithms and experimental results in Appendices C and D.

2. Experiments with Gaussian Data

In order to develop our findings clearly, in this section, we adopt a widely used Gaussian input model (Lee et al., 2018). By examining this well-defined Gaussian model for classification, we establish the evidence for a classification difficulty driving regularization via the confirmation of a transition phenomenon in the spectra of network's weight matrices in the order of HT $\rightarrow$ BT $\rightarrow$ LT. Moreover, the transition is quantitatively controlled by (i) the SNR of the Gaussian model, and (ii) the number of classes K in the model[2].

Empirically Results: Signal-to-noise ratio (SNR) is a common indicator to measure data quality and greatly impacts the classification difficulty in a Gaussian model. We empirically examine the spectra by changing the SNR and the number of classes K in different architectures:

1. Different NN structures: wider but shallower, or narrower but deeper. These structures are similar to the various well known NNs' fully connected denser layers, such as LeNet and MiniAlexNet;

2. Different layers in neural networks: all weight matrices in different layers have spectrum transition driven by the SNR and the number of classes K;

3. Different class numbers in input data: the spectrum transition is always observed in different class numbers, and HT is more likely to emerge when increasing the number of classes K.

Table 1 gives a short summary of the findings when changing the SNR.

We empirically observe the spectrum transition in all settings. The transition is fully driven by the classification difficulty. Therefore, in this Gaussian model, the indicated implicit regularization in the trained DNN is data-effective, directly determined by the difficulty. Precisely, under low level SNR or high class numbers, the weight matrices of a DNN deviate far away from the common MP model. Instead, they are connected to very different random matrix models. The decrease of classification difficulty drives the weight matrices from Heavy Tailed model into MP models at the final training epoch.

2. Codes are given in https://github.com/juve-xx/watchtheweight

Table 1: Summary of spectrum transition in a controlled Gaussian model with K classes and various SNRs.

SNR	Type of spectra	Number of spikes
Weak	Heavy Tail	$K-1$ or K
Middle	Heavy Tail ↓ Bulk Transition period	$K-1$ or K
Strong	Light Tail (MP Law)	$K-1$ or K

2.1 Gaussian Data Sets

For the multi-classification task, Gaussian model is a commonly used model for assessing theoretical properties of a learning system (Lee et al., 2018). In this model with K classes, data from a class $k \in \{1, \ldots, K\}$ are p-dimensional vector of the form

$$h_{i,k} = \mu_k + \varepsilon_{i,k}, \quad 1 \leq i \leq n_k, \tag{2.1}$$

where $\mu_k \in \mathbb{R}^p$ is the class mean, $\varepsilon_{i,k} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2 I_p)$ are Gaussian noise, n_k is the total number of observation from class k. (This Gaussian data model is referred to as the K-way ANOVA model in the statistics literature.) The signal-to-noise ratio (SNR) for this K-class Gaussian model is defined as

$$\text{SNR} = \underset{\{k,k'\}}{\text{Ave}} \frac{||\mu_k - \mu_{k'}||}{\sigma}. \tag{2.2}$$

Here $||\cdot||$ denotes the Euclidean norm in $\mathbb{R}^p$, and the average is taken over the $\binom{K}{2}$ pairs of classes.

We aim at examining the impact of the classification difficulty on the weight matrix spectra in a trained NN for such Gaussian data. We thus consider two settings for the class means $\{\mu_k\}$ which lead to different families of SNRs. In all the remaining discussions, we will take $\sigma = 1$.

DATASET $\mathcal{D}_1(\delta)$: CLASS MEANS WITH RANDOMLY SHUFFLED LOCATIONS

Consider a base mean vector $u = (m, \ldots, m, m+\delta, \ldots, m+\delta)^T \in \mathbb{R}^p$ where half of the components are m, and the other half, $m+\delta$. For the class means μ_k, we reshuffle the locations of these components randomly (and independently). Formally, for each class k, we pick a random subset $I_k \subset \{1, ..., p\}$, of size $p/2$, and define the mean for this class as

$$\mu_k = m\mathbf{1}_{I_k} + (m+\delta)\mathbf{1}_{I_k^c}. \tag{2.3}$$

Here for a subset $A \subset \{1, ..., p\}$, $\mathbf{1}_A$ is the indicator vector of A with coordinates $\mathbf{1}_A(i) = \mathbf{1}_{\{i \in A\}}$ $(1 \leq i \leq p)$.

This setting with randomized locations is motivated by an essential empirical finding from exploring a few classical trained DNNs such as MiniAlexNet and LeNet. Indeed, we found that in these DNNs, the global histograms of the features from all the neurons are pretty similar, with very comparable means and variances, for various NNs; the differences across the NNs are that high and low values of the features appear in different neurons (locations). The randomly shuffled means

used in our experiments are designed to imitate these working mechanisms observed in real-world NNs.

It follows that for the difference $\mu_k - \mu_{k'} = (z_j),\ 1 \le j \le p$ from two classes $k \ne k'$, its coordinates z_j take on the values $-\delta$, 0 and δ with probability $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{1}{4}$, respectively. Clearly, the model SNR will depend on the tuning parameter δ. By Hoeffding inequality, we first conclude that

$$P\left(\left|\frac{||\mu_k - \mu_{k'}||^2}{p} - \frac{\delta^2}{2}\right| \le \epsilon\delta^2\right) \ge 1 - \exp\left(-2\epsilon^2 p\right),$$

or equivalently,

$$P\left(\frac{\delta}{\sqrt{2}}\sqrt{1-2\epsilon} \le \frac{||\mu_k - \mu_{k'}||}{\sqrt{p}} \le \frac{\delta}{\sqrt{2}}\sqrt{1+2\epsilon}\right) \ge 1 - \exp\left(-2\epsilon^2 p\right)$$

Note that $\sqrt{1+x} \le 1 + x$, $\sqrt{1-x} \ge 1 - x$ when $0 < x < 1$. By taking $\epsilon = \sqrt{\log p/p}$, we conclude that with probability at least $1 - 1/p^2$,

$$\left|||\mu_k - \mu_{k'}|| - \delta\sqrt{\frac{p}{2}}\right| \le \delta\sqrt{2\log p}.$$

Therefore at a first-order approximation, the SNR (2.2) in this Gaussian model is (with $\sigma = 1$),

$$\text{SNR} = \underset{\{k,k'\}}{\text{Ave}} \frac{||\mu_k - \mu_{k'}||}{\sigma} \sim \delta\sqrt{\frac{p}{2}}. \tag{2.4}$$

DATASET $\mathcal{D}_2(t)$: CLASS MEANS OF ETF TYPE

Consider the family of vectors $\{v_k\}_{1\le k\le K}$ where v_k is defined by

$$v_k = \mathbf{1}_{\{i=k\}} - \frac{1}{K}\mathbf{1}_{\{1\le i\le K\}}, \quad 1 \le i \le p.$$

So v_k has support on $\{1, \ldots, K\}$ and $||v_k|| = \sqrt{(K-1)/K}$. The normalized family $\{v_k/||v_k||\}$ is called a K-standard ETF structure (Papyan et al., 2020).

We define the k-th class mean as $\mu_k = tv_k$, and use the scale parameter $t > 0$ to tune the SNR of the model. It is easy to see that $||\mu_k - \mu_{k'}|| = \sqrt{2}t$ so that the model SNR is

$$\text{SNR} = \underset{\{k,k'\}}{\text{Ave}} \frac{||\mu_k - \mu_{k'}||}{\sigma} = ||\mu_k - \mu_{k'}|| = \sqrt{2}t. \tag{2.5}$$

(Papyan et al., 2020) has shown that the ETF structure is an optimal position for the final training outputs. Many experiments on real data sets lead to ETF structure for final engineered features. From a layer-peered perspective as mentioned in (Ji et al., 2021), each layer in NN can be regarded as an essential part of feature engineering, and the feature is extracted layer by layer. The ETF structure model considers that the first Dense layer behind the convolution layer is already close to the end of feature extraction.

In our experiments, we take $m = -0.2$ (and $\sigma = 1$). The size of each class k is $n_k = 7500$ in the training dataset, and $n_k = 800$ in test dataset. The number of classes K takes on the values $\{2, 5, 8\}$ on all datasets. Table 2 gives the ranges of the model SNR observed in different dataset/NN combinations with the chosen values of tuning parameters δ and t.

Table 2: Range of SNRs observed in various datasets/ networks combinations.

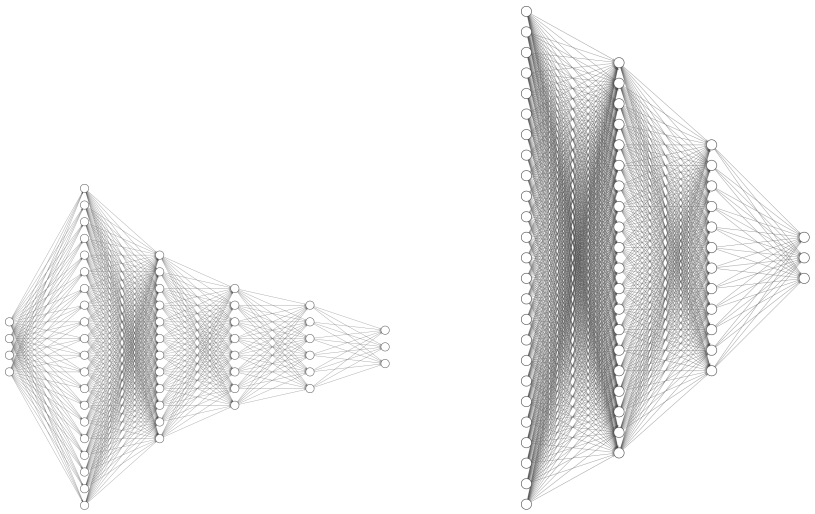
	$\mathcal{D}_1(\delta)$		$\mathcal{D}_2(t)$	
	δ	SNR interval	t	SNR interval
NN1	0.01	[0.01, 1.19]	0.08	[0.08, 4.80]
	0.05	[1.20, 2.00]		
NN2	0.005	[0.005, 0.4]	0.08	[0.08, 4.80]

2.1.1 STRUCTURE OF NEURAL NETWORKS

We consider two different neural networks, a narrower but deeper NN1, and a wider but shallower NN2. The number of layers and their dimensions are shown in Figure 1:

NN1: $100 \to 1024 \to 512 \to 384 \to 192 \to K$,
NN2: $2048 \to 1024 \to 512 \to K$.

The activation function is $\text{ReLU}(x) = \max(x, 0)$. We do not apply any activation function on the last layer.



(a) NN1 (b) NN2

Figure 1: The two NNs considered which imitate the dense layer in well-known NNs such as MiniAlexNet, VGG and LeNet.

2.1.2 OPTIMIZATION METHODOLOGY

Following common practice, we minimize the cross-entropy loss using stochastic gradient descend with momentum 0.9. All the datasets are trained with batch size =64 on a single GPU, for 248 epochs. Trained NNs are saved for the first 10, and then every four epochs. The total number of saved NNs is $(136 + 60 + 80 + 60) \times 3 \times 70 = 70560$. The initialization is Pytorch's default initialization, which follows a uniform distribution. The learning rate is 0.01.

2.2 Results on synthetic data experiments

To investigate the influence of the data SNR on the whole training process, we first report synthetic data experiments results.

2.2.1 THREE TYPES OF SPECTRUM BULK

We use SNR to measure the data quality and focus on the non-zero eigenvalues of the matrix WW^T. Clearly the SNR can directly impact on the classification difficulty. The weight matrices W we consider in this section are those at the final epoch (248th). In the Gaussian data sets, with different values of SNR, we have observed the following three typical types for the bulk spectrum of the weight matrices:

HT : Heavy Tail

BT : Bulk Transition

LT : Light Tail (MP Law)

We gradually increase the SNR of the Gaussian model and report in Figures 2-4 examples of spectra of weight matrices at the end of training. The SNR is increasing from Figure 2 to Figure 4 and within each figure, from plot (a) to plot (d). In Figure 2 the SNR is relatively low, the weight matrix spectra (in blue) show significant departure from the reference MP spectrum (in red). These spectra are defined as of heavy tail type (class HT). In contrast, spectra in Figure 4 with relatively high SNR, closely match the reference MP spectrum, and this corresponds to the light tail class LT. More complex structures appear in the intermediate Figure 3 which correspond to medium values of the SNR. A transition is taking place from Figure 3(a), which is still close to a HT spectrum, to Figure 3(d), which is now close to a MP spectrum. Spectra as those shown in Figue 3 are referred as the bulk transition class BT.

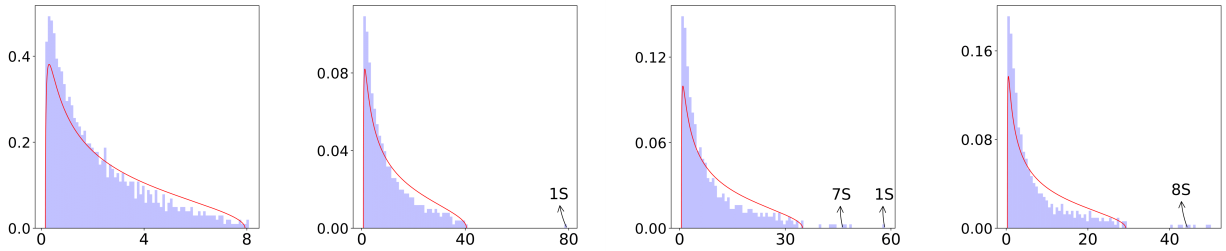
In addition to the bulk transition above, the spike eigenvalues (outliers) also have a characteristic movement. Papyan (2020) reported that in general the total $K = 8$ spikes are grouped in two clusters with $K-1 = 7$ spikes (determined by the between-class covariance matrix) and a singleton (determined by the general mean), respectively. We now describe the evolution of the group and the singleton with gradually increased SNR and the full transition HT$\rightarrow$BT$\rightarrow$LT between the bulk classes. At the very beginning (Figure 2(a)), all the spikes are hidden in the bulk. When the SNR increases, the group of 7 spikes emerge from the bulk and stay outside the spectrum forever. The movement of the singleton spike is more complex, hiding in and leaving the bulk repeatedly. There are particular moments where the group and the singleton meet and stay close each other: we then see a group of 8 spikes.

We use "XX(m,n)" to describe the whole empirical spectral distribution (ESD) of weight matrices including both the bulk and spikes. Here "XX" means one of the three bulk types in $\{$HT, BT, LT$\}$. The number "m" or "n" gives us position information of the two groups of the spikes, numbered in increasing order of their values. For instance, BT(1,7) displayed in Figure 3(d), means the bulk type is BT, the singleton spike lays between the bulk and the group of 7 spikes; HT(0,8) means the group of 7 spikes and the singleton are mixed; HT(0,7) means we see only the group of 7 spikes.

Remark 1 The spectrum transition from HT to BT and LT can also be assessed by more quantitative criteria. (i) The transition from HT to BT is related to the position of the group of $K-1$ spikes, the singleton spike and the bulk edge. When the group of $K-1$ spikes is large enough, the HT type

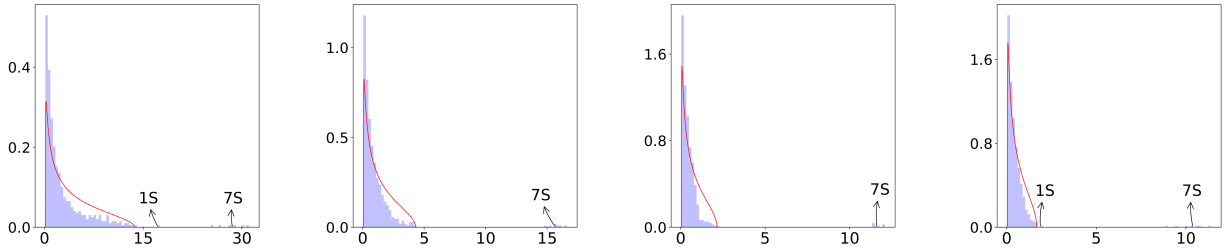
ends and the BT phase starts. Note that here the bulk type is heavy-tailed in both regimes HT and BT. (ii) The transition from BT to LT can be directly detected by comparing the bulk spectrum to the reference MP spectrum. Precisely, this can be achieved using our spectral distance statistic $\hat{s}_n$ introduced in Section 4.

Remark 2 Regarding the special case of the MP spectrum with unit aspect ratio, the density is unbounded at the origin. However, the right edge is regular and the spectrum is still classified as a LT type.



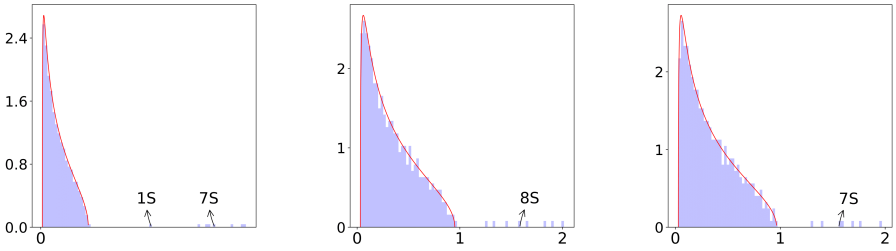
(a) HT(0,0) (b) HT(0,1) (c) HT(7,1) (d) HT(0,8)

Figure 2: Examples of observed HT type spectrum bulks. From plot (a) to (d) the SNR increases. The experiments are conducted from Synthetic data and the pictures are examples for the specific classification.



(a) BT(1,7) (b) BT(0,7) (c) BT(0,7) (d) BT(1,7)

Figure 3: Examples of observed Bulk Transition (BT) type spectrum bulks. From plot (a) to (d) the SNR increases from the first and second column to the third and last. The experiments are conducted from Synthetic data and the pictures are examples for the specific classification.



(a) LT(1,7) (b) LT(0,8) (c) LT(0,7)

Figure 4: Examples of observed LT type spectrum bulks. From plot (a) to (c) the SNR increases from LT(1,7) to LT(0,8) and LT(0,7). The experiments are conducted from Synthetic data and the pictures are examples for the specific classification.

Rank Collapse: One special case, Rank Collapse, occasionally emerges in our experiments especially when SNR is low. This is the phenomenon that some spike eigenvalue is huge, making the bulk in the picture 'needle like' as shown in Figure 5. When the classification difficulty decreases (SNR increases), Rank Collapse gradually disappears.

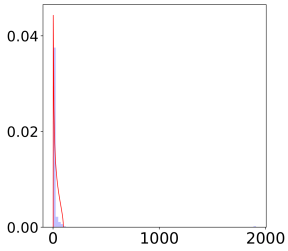


Figure 5: Example of spectrum with Rank Collapse. The figure here is an example to display the rank collapse.

In Martin and Mahoney (2021b), the 5+1 phases of training correspond to the 5 phases of Random Matrix Theory that arise when considering both Gaussian as well as Heavy Tailed (HT) random matrices, and which include 3 different HT phases. In this work, they propose to identify which of the 3 HT phases an HT layer is by fitting the spectral density to a power law, and considering fitted the PL exponent alpha; the value of alpha determines the universality class. Using this classification, they have discovered that most very well trained DNNs have layer that live in what they refer to as the Fat Tailed, or Moderately Heavy Tailed (MHT), Universality class (with layer Power Law exponents α between 2 and 6).

Here, we reformulate their approach, and propose a smaller classification, that includes only 1, not 3, HT phases[3]. Our single HT phase is not identified with any particular HT Universality class of RMT, nor do we attempt to identify or fit the specific HT distribution (i.e Power Law, Truncated Power Law, etc). Instead, we propose a quantitative but non-parametric spectral criteria to identify the appearance of our HT phase. And we note that, using this spectral criteria, the onset of our HT phase corresponds to a good stopping criteria, indicating when a model is well trained[4].

2.2.2 PHASE TRANSITION

We now provide detailed evidence that the spectrum bulks of weight matrices undergo a phase transition controlled by the data SNR. As mentoned earlier, the phase transition operates in the direction of

$$\text{HT} \rightarrow \text{BT} \rightarrow \text{LT}$$

when the SNR increases. The complete experimental results, with recorded phase transition periods (in terms of intervals of SNR values) in all NN layers, are given in Tables 3-4, for the four NN/dataset combinations respectively. These tables are summarized in Figure 6 as a graphical summary.

The main findings from these results are as follows:

1. For all the four NN/dataset combinations, all the three tested class numbers K and in all layers, the ESDs always observe the **same phase transition direction**: HT→BT→LT.

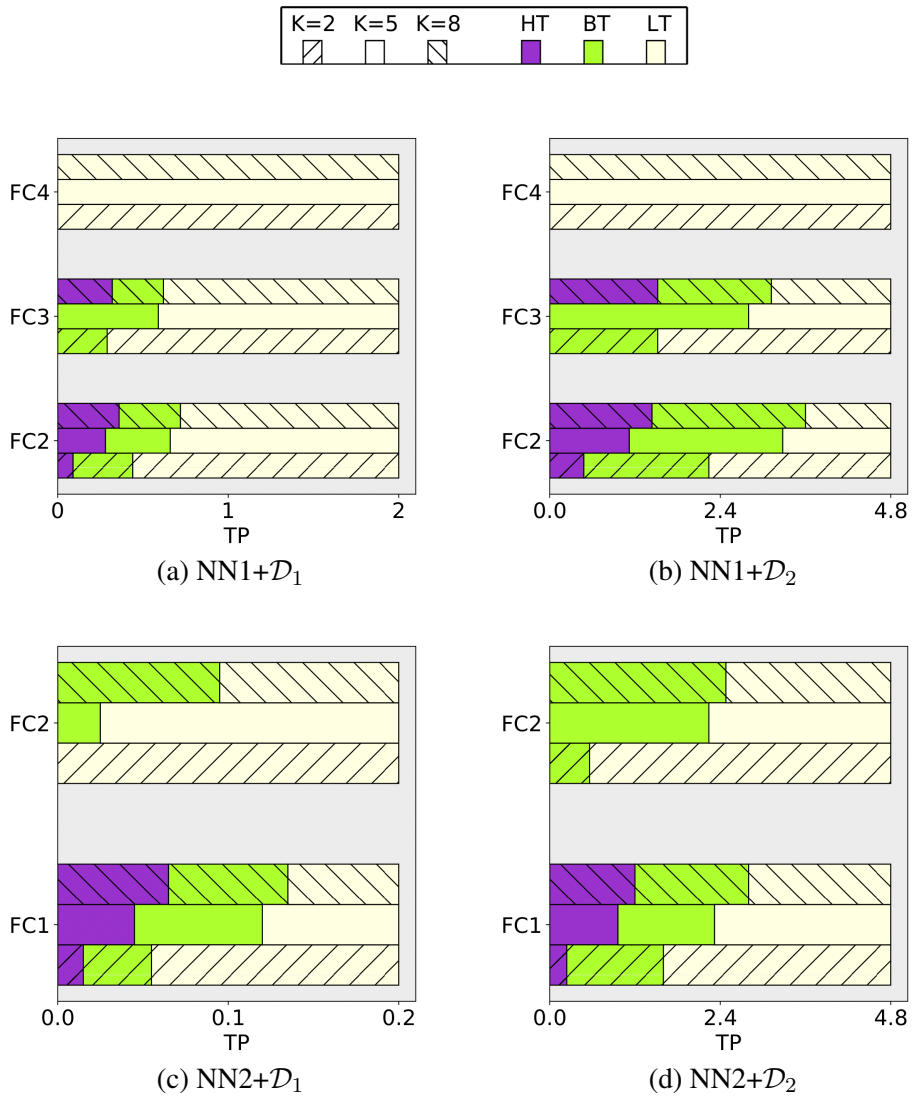
   When the data SNR is high enough, all spectrum bulks in weight matrices fall into the LT type. It is also noted that some weight matrices start from BT type and LT type meaning that the SNR has never been too low. The bulk transition (BT) period kicks off when the group of $K-1$ spikes separate from the bulk.

   In our experiments, these transitions from HT to LT are fully generated by and only responsible to the single tuning parameter, namely the data SNR, or the difficulty of the classification problem. There is thus evidence of a strong impact of the classification difficulty on the weight matrices spectra.

2. For a given layer in the neural network at a same SNR level, HT has higher probability to emerge as the number of classes K increases: the more classes the data set has, the higher difficulty to classify them correctly. The phenomenon that HT emerges with an increased number of classes K thus gives another evidence of strong impact of the classification difficulty on weight matrices spectra.

3. One interesting phenomenon is that when one travels from the initial layer to deeper layers (FC2→FC4 in NN1, FC1→FC2 in NN2), the layers become narrower and the tails of spectrum bulks become lighter. This is true for both NNs and all SNR levels. In line with the previous work in Hodgkinson and Mahoney (2021), the statement that the wider layers will exacerbate HT is validated in our experiments. Practioners are thus suggested to design wider layers for learning process monitoring.

3. We are grateful to a referee who extensively helped us clarify the relationship between our classification and the 5+1 taxonomy of Martin and Mahoney (2021b).
4. It is noted that our spectral criterion resembles the non-parametric weightwatcher rand distance metric, proposed in Yang et al. (2022), available in the open-source weightwatcher package. (https://weightwatcher.ai)



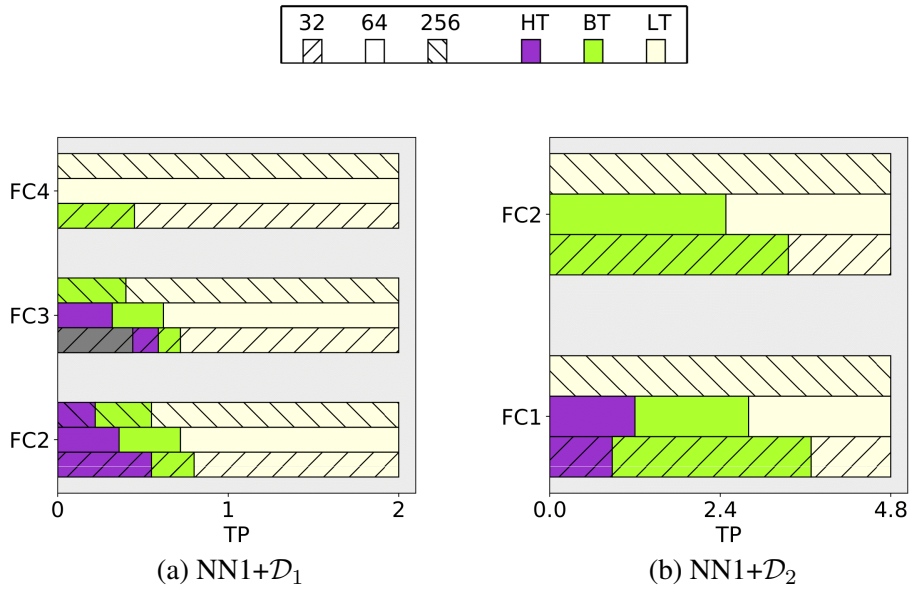
(a) NN1+$\mathcal{D}_1$ (b) NN1+$\mathcal{D}_2$

(c) NN2+$\mathcal{D}_1$ (d) NN2+$\mathcal{D}_2$

Figure 6: Transition Period with the four NN/dataset combinations. The x-axis is the tuning parameter (TP) to tune the SNR level with the range given in Table 2. Each block of three lines in a given layer corresponds to the cases $K = 8$ (Topline), $K = 5$ (Middle line) and $K = 2$ (Bottom line). Different colors represent different spectrum types in {HT, BT, LT}.

2.2.3 ADDITIONAL EXPERIMENTS ON DIFFERENT BATCH SIZES

Batch size also has great impact on the training stability together with data features. (Keskar et al., 2016; Goyal et al., 2017) observe that different batch sizes may give different influence on the training dynamics. To give more comprehensive evidence of the impact of classification difficulty on weight matrices spectra, we now conduct experiments on NN1+$\mathcal{D}_1$ and NN2+$\mathcal{D}_2$ with $K = 8$, and two additional batch sizes $256, 32$ (previous experiments all used a batch size of 64). The other settings are identical as in the previous experiments with varying SNRs.



(a) NN1+$\mathcal{D}_1$ (b) NN1+$\mathcal{D}_2$

Figure 7: Transition Period in different batch sizes. The gray part in NN1+$\mathcal{D}_1$ at FC3 represents Rank Collapse.

The results are fully reported in Table 5 while a graphic sketch in given in Figure 7. In each row on the figure and from left to right, the SNR is increasing, that is the classification difficulty is decreasing, it is amazing to see that the same phase transition of the bulk spectrum still takes place here and in the *same direction* as previously (HT→BT→LT), for all batch sizes.

It is also interesting to look at the figure vertically, from bottom to top, the batch size is increasing from 32 to 256. We observe again the transition in the direction HT→BT→LT. Actually, this phenomenon of observing more likely HT with smaller batch sizes has already been reported in Martin and Mahoney (2021b), so confirmed by our new experiments.

3. Experiments with Real Data

The previous results are based on synthetic Gaussian data. Here we conduct experiments with real data sets to show the impact of classification complexity/difficulty on weight matrices spectra. The DNNs chosen for these experiments are LeNet, MiniAlexNet and VGG11 (LeCun et al., 1998; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014), which are the most classic and representative DNNs in pattern recognition. We consider two data sets, the MNIST and CIFAR10. Note

Table 3: The results of ESD types in different layers with different SNR in **NN1**.

$K = 2$ (NN1+$\mathcal{D}_1$)

FC2		FC3		FC4	
[0.01,0.03]	HT(0,1)				
[0.04,0.09]	HT(1,1)	-		-	
[0.1,0.44]	BT(1,1)	[0.01,0.29]	BT(1,1)	-	
[0.45,2]	LT(1,1)	[0.3,0.38]	LT(1,1)	[0.01,2]	LT(0,1)
		[0.39,2]	LT(0,1)		

$K = 2$ (NN1+$\mathcal{D}_2$)

FC2		FC3		FC4	
[0.08,0.24]	HT(0,1)				
[0.32,0.4]	HT(1,1)	-		-	
[0.48,2.16]	BT[1,1]	[0.08,1.44]	BT[1,1]	-	
[2.24,4.8]	LT(1,1)	[1.52,2.64]	LT(1,1)	[0.48,4.8]	LT(0,1)
		[2.72,4.8]	LT(0,1)		

$K = 5$ (NN1+$\mathcal{D}_1$)

FC2		FC3		FC4	
[0.01,0.15]	HT(0,1)				
[0.16,0.2]	HT(4,1)	-		-	
[0.21,0.28]	HT(0,5)				
[0.29,0.66]	BT(1,4)	[0.01,0.59]	BT(1,4)	-	
[0.67,1.04]	LT(1,4)	[0.6,1.9]	LT(1,4)	[0.01,2]	LT(0,4)
[1.05,2]	LT(0,5)	[1.95,2]	LT(0,5)		

$K = 5$ (NN1+$\mathcal{D}_2$)

FC2		FC3		FC4	
[0.08,0.48]	HT(0,1)				
[0.56,0.64]	HT(4,1)	-		-	
[0.72,1.04]	HT(0,5)				
[1.12,3.2]	BT(1,4)	[0.08,2.72]	BT(1,4)	-	
[3.28,4.8]	LT(1,4)	[2.8,4.8]	LT(1,4)	[0.08,4.8]	LT(0,4)

$K = 8$ (NN1+$\mathcal{D}_1$)

FC2		FC3		FC4	
[0.01,0.19]	HT(0,1)				
[0.2,0.25]	HT(7,1)	[0.01,0.32]	HT(0,8)	-	
[0.26,0.36]	HT(0,8)				
[0.37,0.72]	BT(1,7)	[0.32,0.63]	BT(1,7)	-	
[0.73,1.01]	LT(1,7)	[0.64,1.15]	LT(1,7)		
[1.02,1.9]	LT(0,8)	[1.16,1.95]	LT(0,8)	[0.01,2]	LT(0,7)
[1.95,2]	LT(0,7)	[2,2]	LT(0,7)		

$K = 8$ (NN1+$\mathcal{D}_2$)

FC2		FC3		FC4	
[0.08,0.56]	HT(0,1)	[0.08,1.12]	HT(7,1)		
[0.64,0.96]	HT(7,1)	[1.2,1.44]	HT(0,8)	-	
[1.04,1.36]	HT(0,8)				
[1.44,3.52]	BT(1,7)	[1.52,3.12]	BT(1,7)	-	
[3.6,4.64]	LT(1,7)	[3.2,4.8]	LT(1,7)	[0.08,4.8]	LT(0,7)
[4.72,4.8]	LT(0,8)				

* The interval is the range of tuning parameters which tune data SNR.

Table 4: The results of ESD types in different layers with different SNR in **NN2**.

$K = 2$ (NN2+$\mathcal{D}_1$)

FC1		FC2	
[0.005,0.015]	HT(0,1)		
		-	
[0.02,0.055]	BT(1,1)	-	
[0.06,0.235]	LT(1,1)	[0.005,0.4]	LT(1,1)
[0.24,0.4]	LT(0,1)		

$K = 2$ (NN2+$\mathcal{D}_2$)

FC1		FC2	
[0.08,0.16]	HT(0,1)		
		-	
[0.24,1.52]	BT[1,1]	[0.08,0.48]	BT[1,1]
[1.6,1.92]	LT(1,1)	[0.56,1.6]	LT(1,1)
[2,4.8]	LT(0,1)	[1.68,4.8]	LT(0,1)

$K = 5$ (NN2+$\mathcal{D}_1$)

FC1		FC2	
[0.005,0.015]	HT(0,1)		
[0.02,0.035]	HT(4,1)	-	
[0.04,0.045]	HT(0,5)		
[0.05,0.12]	BT(1,4)	[0.005,0.125]	BT(1,4)
[0.125,0.4]	LT(1,4)	[0.03,0.38]	LT(1,4)
		[0.385,0.4]	LT(0,5)

$K = 5$ (NN2+$\mathcal{D}_2$)

FC1		FC2	
[0.08,0.48]	HT(0,0)		
[0.56,0.56]]	HT(0,1)	-	
[0.64,0.88]	HT(0,5)		
[0.96,2.24]	BT(1,4)	[0.08,2.16]	BT(0,4)→BT(1,4)
[2.32,2.48]	LT(1,4)	[2.24,4.8]	LT(1,4)
[2.56,4.8]	LT(0,4)		

$K = 8$ (NN2+$\mathcal{D}_1$)

FC1		FC2	
[0.005,0.025]	HT(0,1)		
[0.03,0.06]	HT(7,1)	-	
[0.065,0.065]	HT(0,8)		
[0.07,0.135]	BT(1,7)	[0.005,0.095]	BT(1,7)
[0.14,0.4]	LT(1,7)	[0.1,0.24]	LT(1,7)
		[0.245,0.4]	LT(0,8)

$K = 8$ (NN2+$\mathcal{D}_2$)

FC1		FC2	
[0.08,0.8]	HT(0,0)		
[0.88,0.96]	HT(0,1)	-	
[1.04,1.12]	HT(0,8)		
[1.2,2.72]	BT(1,7)	[0.08,2.4]	BT(1,7)
[2.8,3.04]	LT(1,7)	[2.48,4.64]	LT(1,7)
[3.12,4.8]	LT(0,7)	[4.72,4.8]	LT(0,8)

* The interval is the range of tuning parameters which tune data SNR.

Table 5: The results of ESD types in different layers with different SNR in different batchsizes.

Batchsize= 32 (NN1+$\mathcal{D}_1$)

FC2		FC3		FC4	
[0.01,0.17]	HT(0,1)	[0.01,0.44]	RC		
[0.18,0.46]	HT(7,1)	[0.45,0.52]	HT(7,1)	-	
[0.47,0.55]	HT(0,8)	[0.52,0.59]	HT(0,8)		
[0.56,0.84]	BT(1,7)	[0.6,0.78]	BT(1,7)	[0.15,0.45]	HT(0,7)
[0.85,1.08]	LT(1,7)	[0.79,1.12]	LT(1,7)	[0.46,2]	LT(0,7)
[1.09,2]	LT(0,8)	[1.13,2]	LT(0,8)		

Batchsize= 32 (NN2+$\mathcal{D}_2$)

FC1		FC2	
[0.08,0.8]	HT(0,0)	-	
[0.88,3.6]	BT(0,7)	[0.08,3.28]	RC→BT(1,7)
[3.68,4.8]	LT(0,7)	[3.36,4.8]	LT(1,7)

Batchsize= 64 (NN1+$\mathcal{D}_1$)

FC2		FC3		FC4	
[0.01,0.19]	HT(0,1)				
[0.2,0.25]	HT(7,1)	[0.01,0.32]	HT(0,8)	-	
[0.26,0.36]	HT(0,8)				
[0.37,0.72]	BT(1,7)	[0.33,0.63]	BT(1,7)	-	
[0.73,1.01]	LT(1,7)	[0.64,1.15]	LT(1,7)	[0.01,2]	LT(0,7)
[1.02,1.9]	LT(0,8)	[1.16,1.95]	LT(0,8)		
[1.95,2]	LT(0,7)	[2,2]	LT(0,7)		

Batchsize= 64 (NN2+$\mathcal{D}_2$)

FC1		FC2	
[0.08,0.8]	HT(0,0)		
[0.88,0.96]	HT(0,1)	-	
[1.04,1.12]	HT(0,8)		
[1.2,2.72]	BT(1,7)	[0.08,2.4]	BT(1,7)
[2.8,3.04]	LT(1,7)	[2.48,4.64]	LT(1,7)
[3.12,4.8]	LT(0,7)	[4.72,4.8]	LT(0,8)

Batchsize= 256 (NN1+$\mathcal{D}_1$)

FC2		FC3		FC4	
[0.01,0.16]	HT(0,0)				
[0.17,0.22]	HT(0,8)	-		-	
[0.23,0.55]	BT(0,7)→BT(1,7)	[0.01,0.4]	BT(1,7)	-	
		[0.41,1.01]	LT(1,7)		
[0.56,2]	LT(1,7)	[1.02,1.95]	LT(0,8)	[0.01,2]	LT(0,7)
		[2,2]	LT(0,7)		

Batchsize= 256 (NN2+$\mathcal{D}_2$)

FC1		FC2	
-		-	
-		-	
[0.08,1.2]	LT(0,0)	[0.08,4.64]	LT(1,7)
[1.28,4.8]	LT(0,7)	[4.72,4.8]	LT(0,8)

* The interval is the range of tuning parameters which tune data SNR.

that in Martin and Mahoney's work, the data sets such as CIFAR10, CIFAR100 and Image1000 all cause HT type spectra due to complex features unlike the MNIST data set. In our experiments, we select MiniAlexNet instead of the more extensive AlexNet to reduce computing complexity.

3.1 Experimental Design

The structures of LeNet and MiniAlexNet are shown in Figure 8, and for VGG we refer the reader to Simonyan and Zisserman (2014). The data sets we use are MNIST and CIFAR10. We tune batch sizes to have different practical architectures, then check spectra type in the trainined NNS with these architectures. As before, we save trained models for the first 10 epochs and every four epochs afterward. The optimization methodology is the same as previously introduced Section 2.1.2.

We concentrate the discussion on the first fully connected layer just next to convolution layers in each NN. The weight matrix has the structure 2450×500 in LeNet, 4096×384 in MiniAlexNet and 2048×500 in VGG.

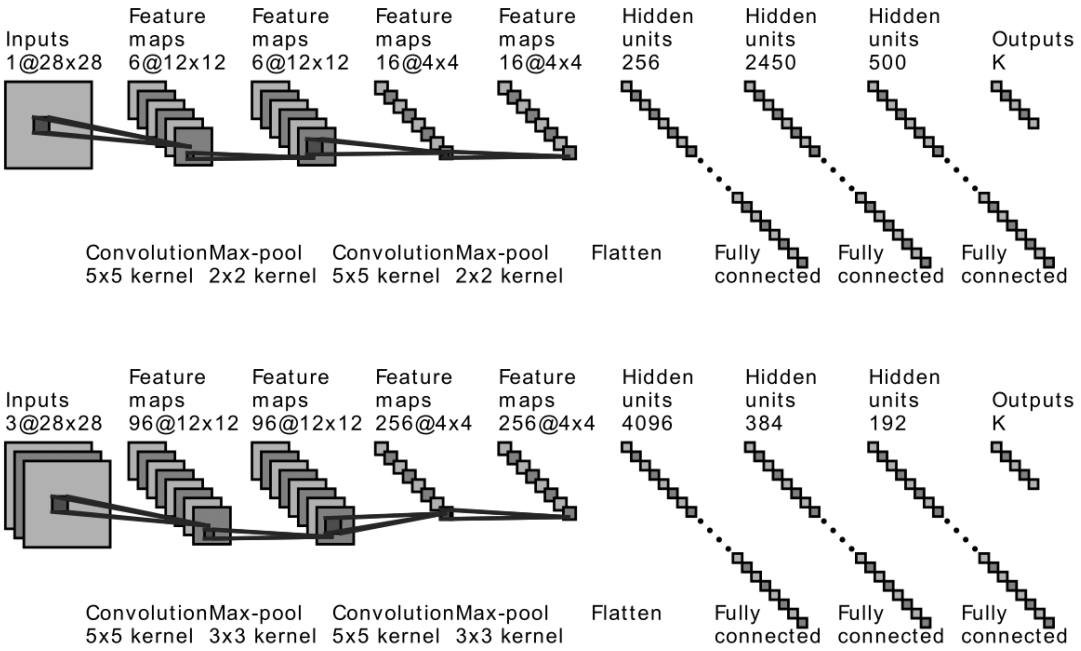


Figure 8: The structure of LeNet (top) and MiniAlexNet (Bottom). The input data sets are MNIST with size $1 \times 28 \times 28$ or CIFAR10 with size $3 \times 28 \times 28$, the fully connected layers we consider lay behind convolutional layers.

3.2 Results

For the MNIST data, Figures 9 reports all the ESDs obtained at the fully connected layer and at the final epoch 248 with various batch sizes within the three NNs. The corresponding results for the

CIFAR10 data are reported in Figures 10. The determination of the spectrum type is done using the method introduced in Section 2.2.1. Comparing the results for a given NN, we see that the spectrum type from the two data sets are consistently different. Specifically,

1. In LeNet+MNIST, the spectra are of BT type except for the batch size 256 with LT type, these BT type spectra are though very close to the MP Law. In LeNet+CIFAR10, the spectra are also of BT type but much closer to the HT type except for the batch size 16;

2. In MiniAlexNet+MNIST, the spectra are of LT type except the batch sizes 16 and 32 with BT type. In MiniAlexNet+CIFAR10, the spectra are of HT type or BT type, all very different from the MP Law;

3. In VGG+MNIST, the spectra are all of LT type; In VGG+CIFAR10, the spectra transit from BT (visibly similar to HT) to LT with increasing batch sizes.

In summary, these experiments show that the spectrum type of the weight matrix depends much more on the data set itself than the NN architecture.

In order to further understand the difference between the two data sets and their impact on the spectrum type of the weight matrices, we evaluate the detection rates on the test data of the trained NNs with the three architectures, see Table 6. The detection rates on MNIST are 99% in all networks, while those on CIFAR10 are 64% with LeNet, 76% with MiniAlexNet and 81% with VGG11, respectively. The differences in testing accuracy give evidence that CIFAR10 has much more complex features than MNIST, and the classification problem is more difficult for CIFAR10 than for MNIST. As the experiments show that training on CIFAR10 is more likely to cause HT spectra, we thus have a novel confirmation that the classification difficulty or complexity has a significant impact on the type of weight matrix spectra. In a sense, complex features in a data set will bring in complex correlations in weight matrix entries, thus generating heavy tails in their spectrum.[5]

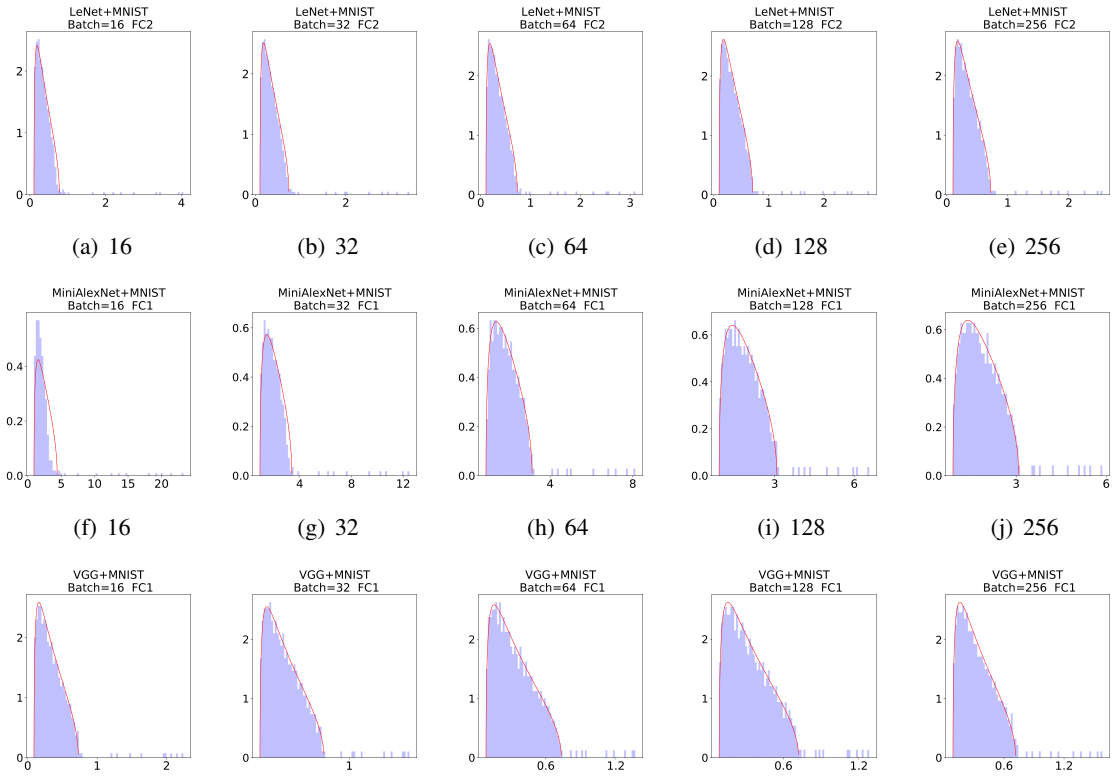
Table 6: Detection rates on test data from the trained NNs with their spectrum type in parenthesis. The batch size is 16.

Data Set \ NN	LeNet	MiniAlexNet	VGG11
MNIST	99% (BT/LT)	99% (BT/LT)	99% (LT)
CIFAR10	64% (HT)	76% (HT)	81% (HT)

4. A spectral criterion for early stopping

As a regularization technology in Deep Learning, early stopping is adopted to improve generalization accuracy of a DNN. People may use testing data set to obtain convenient stopping time

5. Following recommendation from referee, we have selected a few weight matrices in experiments with synthetic Gaussian data which display a HT spectrum and examined their spectrum after reshuffling randomly their entries. We have always observed a LT spectrum after shuffling. This confirms that the HT type spectra found here originated from high correlations between the entries of weight matrices, and not because they have high moments (high values).

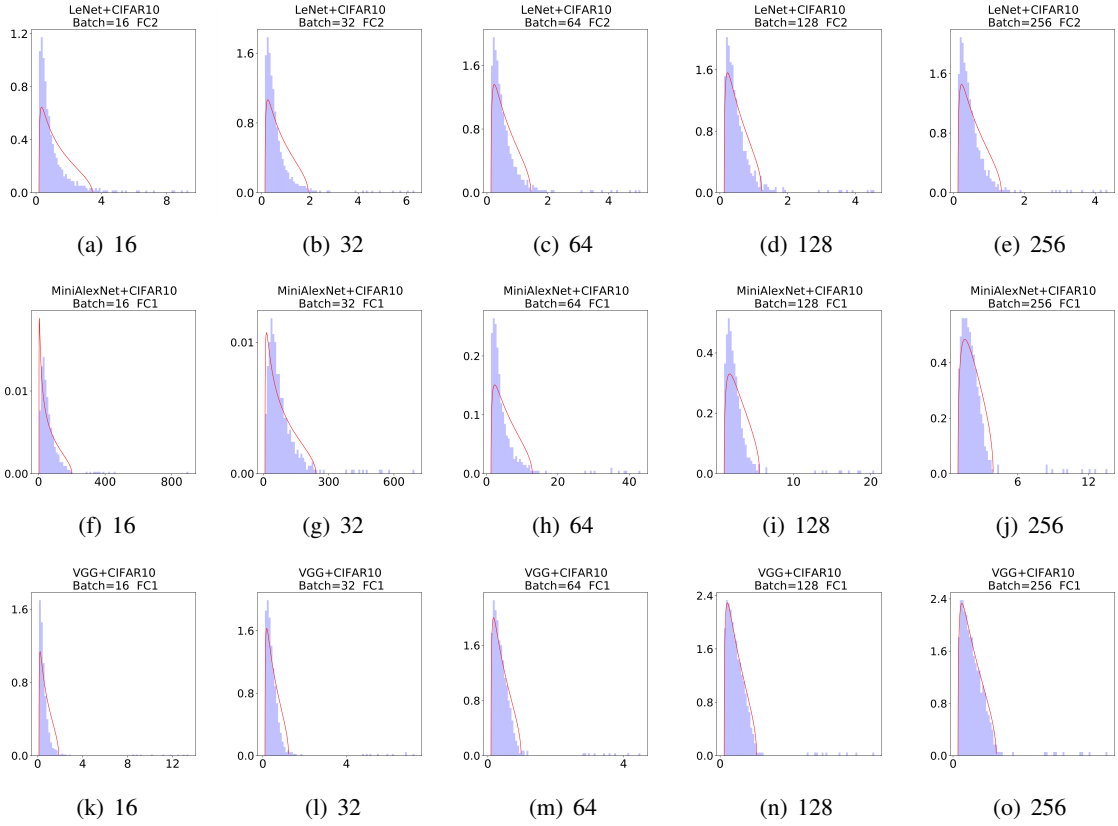


(a) 16 (b) 32 (c) 64 (d) 128 (e) 256

(f) 16 (g) 32 (h) 64 (i) 128 (j) 256

(k) 16 (l) 32 (m) 64 (n) 128 (o) 256

Figure 9: Training on MNIST: Weight matrix spectra at final epoch 248. LeNet: (a)-(e); MiniAlexNet: (f)-(j); VGG : (k)-(o). Columns show experiments with different batch sizes. All the figures are results trained on MNIST. The display of LT in the figures indicates the low classification difficulty of the training on MNIST.



(a) 16 (b) 32 (c) 64 (d) 128 (e) 256

(f) 16 (g) 32 (h) 64 (i) 128 (j) 256

(k) 16 (l) 32 (m) 64 (n) 128 (o) 256

Figure 10: Training on CIFAR10: Weight matrix spectra at final epoch 248. LeNet: (a)-(e); MiniAlexNet: (f)-(j); VGG : (k)-(o). Columns show experiments with different batch sizes. All the figures are results trained on CIFAR10. The display of HT in the figures indicates the complex features in CIFAR10 and high classification difficulty of the training on CIFAR10.

in practice, but when we model the data set, it is a trade off to separate data set into training and testing. Sometimes as Martin et al. (2021) pointed out, it is more expensive to acquire the testing data set. There are also situations where practioners of Deep Learning are laid to use pre-trained and existing DNNs without access to test data.

So an important question we address here is: Without any testing data set, shall we early stop or not? And how to define an early stopping time?

The spectra of weight matrices encode information during the training time and we aim at using this information to guide an eventual early stopping of the training process. Specifically, we construct a *spectral criterion* based on a distance between an ongoing weight matrix spectrum and the reference MP Law. When this distance is judged large enough, we obtain evidence for the formation of a HT or BT type spectrum, thus the implicit regularization in the DNN suggests to stop the training process. Note that this spectral criterion for early stopping does not need any test data.

One may ask **why the appearance of a HT spectrum is a good early stopping time?** To answer the question, recall that Martin et al. (2021) mentioned that MP Law spectra could not evaluate the performance of the trained model, but Heavy Tail spectra could. The Heavy Tail spectra may correspond to better or worse test performance. From the information encoder perspective, we argue that the emergence of Heavy Tail or Rank Collapse in weight matrices could be viewed in two ways:

- Indication of the poor quality in the training data or the poor ability in the whole system: in synthetic data experiments, the poor training data or system quality will lead to instability or overfitting during the training process. So the emergence of Heavy Tail can be treated as an **alarm** for these hidden and problematic issues in the network. Note that this fact of alarm has been also remarked in Martin et al. (2021).

- Indication of a regularized structure that has acquired considerable information from the training data: an HT spectrum is far from the initial MP Laws which is induced by the random weight initialization and its emergence can be viewed as an indication of a well-trained structure that has already captured sufficient information from the input data. Such structure will somehow ensure the testing accuracy of the whole system, and additional training will not bring much improvement.

We now describe this spectral criterion in more detail. Consider a $n \times N$ ($n \leq N$) weight matrix W and let $X_1, X_2, ..., X_n$ be the n non-zero eigenvalues of the square matrix WW^T. (These are also, by definition, the squares of the singular values of the matrix W. The initialization of W has been rescaled with $1/\sqrt{N}$.) We then construct a histogram estimator $\hat{p}_M(x)$ for the joint density of the eigenvalues using M bins. Next, let $p_{c,\sigma^2}(x)$ be the reference Marčenko-Pastur density (Appendix A) depending on a scale parameter σ^2 and a shape parameter $0 < c < 1$, with a compact support $[a, b]$ ($0 < a < b$). In practice, the parameters c and σ^2 in the reference MP density $p_{c,\sigma^2}(x)$ are also estimated by using $X_1, ..., X_n$. This leads to an estimated MP density function $p_{\hat{c},\hat{\sigma}^2}(x)$. The estimation of distance between the distribution of the n eigenvalues and the MP density is defined as the L_1 distance

$$\hat{s}_n = \int_a^b |\hat{p}_M(x) - p_{\hat{c},\hat{\sigma}^2}(x)| dx. \tag{4.1}$$

Under the null hypothesis that the eigenvalues $\{X_i\}$ follow the MP law, we have a precise rate for $\hat{s}_n \to 0$, which leads to our spectral criterion.

Spectral criterion. Set $M = 2\lfloor n^{\frac{1}{3}} \rfloor$ and consider a threshold value $s_* = C * \sqrt{\log n}/n^{\frac{1}{3}}$ with $C = 0.4$. For each training epoch, calculate $\hat{s}_n$ in equation (4.1) by the **Algorithm** in Appendix C. The training is stopped if $\hat{s}_n > s_*$.

(To gain more robustness in this stopping procedure, in all the experiments, we will stop the training at three consecutive epochs where $\hat{s}_n > s_*$ happen (instead of at the first such epoch).) □

More details on the determination of the distance value $\hat{s}_n$ and the threshold value s_* are given later in Section 4.1.

The spectra criterion is validated in both synthetic and real data experiments. Evidence for this **spectral criterion** is developed in details with extensive experimental results in Sections 4.2 and 4.3. In synthetic data experiments, the spectral criterion provides an early stopping epoch where the testing accuracy is much higher than the final testing accuracy, even when the training accuracy is still increasing. In real data experiments, the spectral criterion could also offer high-quality stopping time, ensuring testing accuracy and cutting off a large unnecessary training time.

Note that the idea of using evolution of weight matrices to monitor the training process of a DNN has appeared earlier in the AI community with the online WeightWatcher package. This open-source package allows the user to analyze various pre-trained DNN models, with or without training/testing data. Particularly, it permits the user to estimate the power law exponent α for a given weight matrix and determine in consequence whether the DNN of interest is under-trained or over-trained (Martin and Mahoney, 2021a,b; Martin et al., 2021). Online discussions on this GitHub repository also mentioned how to use WeightWatcher for early stopping. However to our best knowledge, there was no clearly defined implementation of this idea in an academic report. The spectral criterion developed in this paper help fill this gap.

On a different note, it comes to our attention that Yang et al. (2022) proposed a similar distance measure called "randdistance metric" to distinguish between a HT type and MP Law. Among many distance measures considering in the paper, the randdistance is the most recommended one which "achieves the highest worst-case rank correlation with generalization performance under a variety of training settings".

4.1 Technical details of the spectral criterion

Consider n data points $X_1, X_2, ..., X_n$, supported on an interval $[a, b]$, with $0 < a < b$. Consider a mesh net on the interval on M bins of binsize $(b-a)/M$,

$$B_j = \left(a + (j-1)\frac{b-a}{M}, a + j\frac{b-a}{M} \right], \quad 1 \le j \le M.$$

For a real x, let $B(x)$ be the bin B_j that contains x (if no such bin exists, $B(x) = \emptyset$). The histogram estimator for the density function of the data is

$$\hat{p}_M(x) = \frac{M}{n(b-a)} \sum_{i=1}^{n} I(X_i \in B(x)), \quad x \in \mathbb{R}.$$

With reference to Random Matrix Theory Results given in Appendix A, the density function of the MP Law MP_{c,σ^2} is

$$p_{c,\sigma^2}(x) = MP_{c,\sigma^2}(x) = \frac{1}{2\pi c\sigma^2 x}\sqrt{(b-x)(x-a)}I(a \le x \le b), \tag{4.2}$$

with $a = \sigma^2(1-\sqrt{c})^2$ and $b = \sigma^2(1+\sqrt{c})^2$. We thus use the following L_1 distance between the two density functions to measure the departure of the data points $\{X_i\}$ from the MP law:

$$s_n = \int_a^b |\hat{p}_M(x) - p_{c,\sigma^2}(x)|dx. \tag{4.3}$$

Under the null hypothesis that the data points follow the MP-law, we have the following convergence rate of s_n to zero.

Proposition 4.1. *Suppose $\{X_i\}_{i=1}^n$ are generated independently from $p_{c,\sigma^2}(x) = MP_{c,\sigma^2}$, then the distance in* (4.3) *satisfies*

$$s_n = O_p\Big(\frac{1}{M} + \sqrt{\frac{M\log n}{n}}\Big).$$

(Here O_p denotes the boundedness in probability.)

The proof is given in Appendix B.1. Due to the fact that MP density $p_{c,\sigma^2}(x)$ has unbounded derivatives at its edge points $\{a, b\}$, the proof is obtained via a special adaptation of the existing rate for histogram estimator from the literature.

In practice, we do not know the parameters c and σ^2 of the reference MP density $p_{c,\sigma^2}(x)$. Then we use the observed extreme statistics $\hat{a} = X_{(1)}$, and $\hat{b} = X_{(n)}$ to estimate a and b, respectively. These lead to corresponding estimates $\hat{c}$ and $\hat{\sigma}^2$ for the parameters c and σ^2, respectively. The MP density function with estimated parameters is then $p_{\hat{c},\hat{\sigma}^2}(x)$, and the L_1 distance between the data set $\{X_1, \ldots, X_n\}$ and the MP law is estimated by

$$\hat{s}_n = \int_a^b |\hat{p}_M(x) - p_{\hat{c},\hat{\sigma}^2}(x)|dx. \tag{4.4}$$

The following proposition guarantees a convergence rate for the estimator $\hat{s}_n$,

Proposition 4.2. *For the estimated distance $\hat{s}_n$ in* (4.4)*, we have*

$$\hat{s}_n = O_p\Big(\frac{1}{n^{1/3}} + \frac{1}{M} + \sqrt{\frac{M\log n}{n}}\Big). \tag{4.5}$$

The proof of the proposition is given in Appendix B.2. The proposition is next used to define a rejection region for the null hypothesis. Consider $M = O(n^{\frac{1}{3}})$. From (4.5), under the null hypothesis, $\hat{s}_n$ will converge to zero at the optimal rate of $O_P(\sqrt{\log n}/n^{\frac{1}{3}})$. In contrast, under a deviation of ESDs in weight matrices such as emergence of heavy tails, $\hat{s}_n$ will no longer tend to 0. Therefore, it is possible to define a critical value of the form $s_* = C\sqrt{\log n}/n^{1/3}$ for some constant C for the test statistics $\hat{s}_n$. The calibration of the constant C is as follows.

Calibration of the critical constant C. We calibrate the constant C by simulations under the null hypothesis. For different MP Laws, we generate eigenvalues and get histograms of $\hat{s}_n n^{\frac{1}{3}}/\sqrt{\log n}$. As shown in Figure 11, most of the time $\hat{s}_n n^{\frac{1}{3}}/\sqrt{\log n}$ lies in the interval [0.15,0.25] with its largest values around 0.35. We thus recommend the critical constant $C = 0.4$. One may naturally select a slightly different critical constant: we have tested several choices of the constant from $C = 0.4$ to $C = 0.6$ in our experiments and the obtained results are very similar and all satisfactory. Basically, any value of C in the range of $[0.4, 0.6]$ can be recommended for the spectral criterion. □

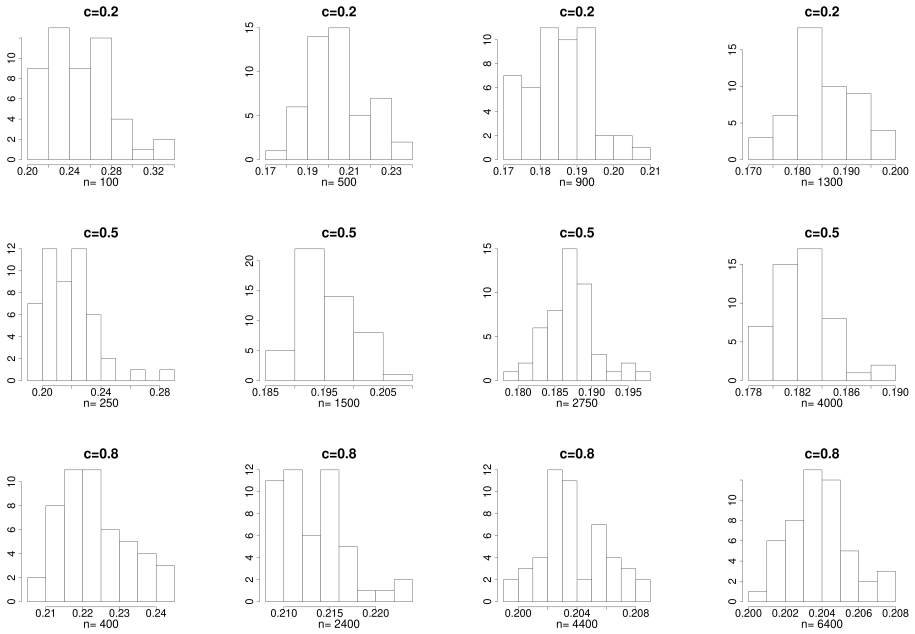


Figure 11: Histograms of $\hat{s}_n n^{\frac{1}{3}}/\sqrt{\log n}$ from different c and n under the null hypothesis: for each pair of (c, n), the eigenvalues are generated from standard MP Law with 50 repetitions that lead to 50 values of the statistic.

4.2 Early stopping in synthetic data experiments

Because of huge amount of data under analysis, we conduct experiments, stock the relevant data, and then check the results offline. The epochs where we save trained NNs for different architectures are all fixed at 0, 1, 2,..., 9, 10, 12, 16, 20,..., 248, the latter epochs having an increment of four. Even in such sparse data reservation, the total data we obtained is larger than 1TB.

We apply the spectra criterion to the stocked training epochs, and decide early stopping if the criterion is met. When this happens, we compare the test accuracy of the stopped NN with that of the NN trained till the final epoch (248th). This comparison serves to measure the quality of the early stopping using the spectral criterion. Experiment results with $K = 8$ are shown in Table 7 and Figure 12.

A summary of findings is as follows.

1. NN1+$\mathcal{D}_1$ and NN1+$\mathcal{D}_2$: During the first 20 epochs when the SNR is low, the testing accuracy is decreasing while the training accuracy is increasing. The spectral criterion detects such hidden and problematic issues and recommends early stopping. It is truly remarkable that almost all early stopped NNs have higher test accuracies than the corresponding NNs trained till the end. The advantage is particularly important when the SNR is low. When the SNR is high, there might be no alarm by the spectral criterion, see the situation of TP=0.9 in NN1+$\mathcal{D}_1$ and TP=4.8 in NN1+$\mathcal{D}_2$ (TP is the tuning parameter reflecting the SNR). This is in fact a consistency of the spectral criterion, no early stopping is needed, and the fully trained NNs have indeed higher test accuracies.

2. NN2+$\mathcal{D}_1$ and NN2+$\mathcal{D}_2$: the spectral criterion detects stopping time under low SNR, nonetheless the testing accuracy is a little lower than the final testing accuracy. As the differences are very small, huge training time is cut off, and testing accuracy is ensured due to the emergence of well-trained structure that already seized sufficient information.

The spectral criterion is also valid when overfitting appears in training. In such situation, the training and testing accuracy do not have the same tendency. As training epochs increase, Figure 12 shows that the training accuracy tends to 100% while the testing accuracy is highly related with the tuning parameter δ or t. Without testing data, the spectral criterion is able to propose an early stopping time even when the training accuracy is still increasing but the testing accuracy becomes to decline.

When the spectral criterion alarms at different epochs in different layers, there is a question that **how to decide a stopping time for the training process?** From our experimental results, we empirically suggest that any epoch after the time some layer hits the critical value s_* is suitable to stop, and it is strongly recommended to stop training if there is more than one layer hitting the critical value. For example, TP=0.15 in NN1+$\mathcal{D}_1$, epochs in 7-10 are all suitable early stopping times with a guaranteed test accuracy.

Table 7: Early stopping results in synthetic data experiments with $C = 0.4$: stopping epochs selected by spectral criterion in different layers' weight matrices and their testing accuracy (Test Acc). The symbol "-" means no early stopping epoch is found by the spectral criterion.

The combination NN1+$\mathcal{D}_1$

Typical TP	spectral criterion $C = 0.4$				Final Epoch 248		
	epoch(FC2)	Test Acc	epoch(FC3)	Test Acc	FC1	FC2	Test Acc
0.15	7	25.84%	10	23.23%	HT	HT	20.17%
0.2	7	32.70%	12	27.48%	HT	HT	27.03%
0.3	7	49.36%	12	45.48%	HT	HT	44.80%
0.6	8	88.52%	32	88.32%	BT	BT	88.30%
0.9	-		-		LT	LT	99.13%

The combination NN1+$\mathcal{D}_2$

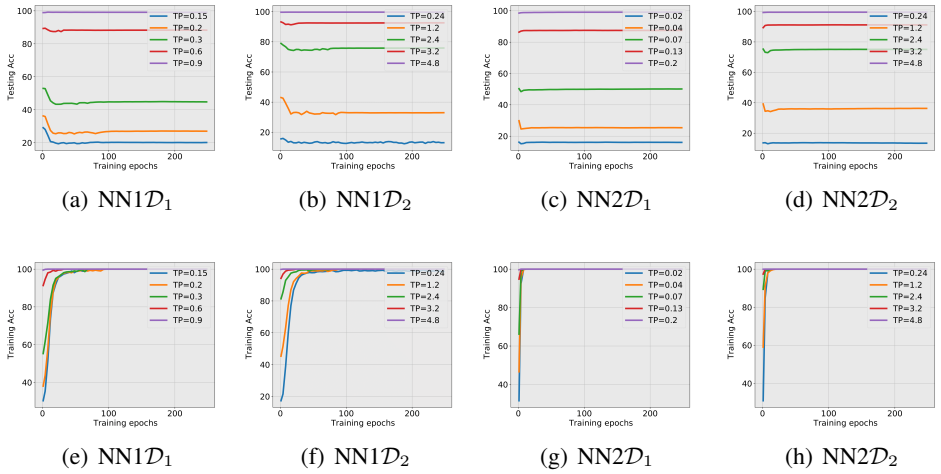
Typical TP	spectral criterion $C = 0.4$				Final Epoch 248		
	epoch(FC2)	Test Acc	epoch(FC3)	Test Acc	FC1	FC2	Test Acc
0.24	9	14.69%	16	13.89%	HT	HT	13.08%
1.2	7	38.61%	12	35.84%	HT	HT	32.98%
2.4	7	77.19%	16	74.55%	BT	BT	75.92%
3.2	9	92.11%	-		BT	LT	92.64%
4.8	-		-		LT	LT	99.73%

The combination NN2+$\mathcal{D}_1$

Typical TP	spectral criterion $C = 0.4$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
0.02	6	14.89%	7	15.84%	HT	BT	16.02%
0.04	8	24.78%	7	23.34%	HT	BT	25.38%
0.07	5	48.31%	6	48.63%	BT	BT	50.12%
0.13	6	87.03%	-		BT	LT	87.50%
0.2	-		-		LT	LT	99.14%

The combination NN2+$\mathcal{D}_2$

Typical TP	spectral criterion $C = 0.4$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
0.24	10	13.08%	6	12.89%	HT	BT	13.44%
1.2	12	34.22%	5	34.63%	BT	BT	36.31%
2.4	5	72.59%	16	74.61%	BT	BT	75.12%
3.2	-		-		LT	LT	91.20%
4.8	-		-		LT	LT	99.59%



(a) NN1$\mathcal{D}_1$ (b) NN1$\mathcal{D}_2$ (c) NN2$\mathcal{D}_1$ (d) NN2$\mathcal{D}_2$

(e) NN1$\mathcal{D}_1$ (f) NN1$\mathcal{D}_2$ (g) NN2$\mathcal{D}_1$ (h) NN2$\mathcal{D}_2$

Figure 12: Testing and Training Accuracy: We begin the line at epoch=1. Testing accuracy: (a)-(d); Training accuracy: (e)-(h). y-axis is the accuracy value, x-axis is the training epochs. Different line represents different SNRs in data sets.

4.3 Early stopping in real data experiments

In real data experiments, we still follow the settings in Section 4.2 to evaluate the quality of early stopping time using the spectral criterion by checking LeNet/MiniAlexNet+MNIST/CIFAR10. Experiment results are shown in Table 8.

A summary of findings is as follows.

1. LeNet+MNIST and LeNet+CIFAR10: Testing accuracy and training accuracy are both increasing during the training process. The FC2 layer in LeNet hits the critical value s_* first, and provides a possible early stopping time. For MNIST, the test accuracy in the early stopped epoch only has the negligible difference 0.1% with the final test accuracy; For CIFAR10, the test accuracy is lower but still guaranteed compared with the final test accuracy. We check the FC1 layer for CIFAR10, find that the "strongly suggested" stopping epochs in batch sizes 16 and 32 have much higher test accuracies, and in larger batch sizes, we could stop for saving time or keep training for a higher test accuracy.

2. MiniAlexNet+MNIST and MiniAlexNet+CIFAR10: The FC1 layer always hits the critical value first. For MNIST, the test accuracy still has negligible difference with the final test accuracy in small batch sizes 16 and 32, and no early stopping epoch is found by the spectral criterion with larger batch sizes. For CIFAR10, it is the most representative experiment because the training explosion happens in batch sizes 16 and 32. We check whether the spectral criterion gives alarm and performs well. The answer is yes. The spectral criterion strongly suggested to stop before training explosion and has a quite high test accuracy. In larger batch sizes, the test accuracy is are also ensured while cutting a large amount of training time.

We also have conducted experiments with another choice of the critical constant of $C = 0.6$ on both synthetic and real data to check the robustness of the spectral criterion. Details are shown in

Appendix D, Tables 9 and 10. The new results are very similar to those obtained with $C = 0.4$. In practice, we recommend the use of the spectral criterion with a critical constant C in the range of [0.4,0.6].

Table 8: Early stopping results in real data experiments with $C = 0.4$: stopping epochs selected by spectral criterion in different layers' weight matrices and their testing accuracy (Test Acc). The symbol "-" means no early stopping epoch is found by the spectral criterion.

The combination LeNet+MNIST

batchsize	spectral criterion $C = 0.4$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
16	-		16	99.08%	LT	BT	99.17%
32	-		40	99.13%	LT	BT	99.17%
64	-		68	98.98%	LT	BT	98.98%
128	-		124	98.91%	LT	BT	99.03%
256	-		-		LT	LT	98.96%

The combination LeNet+CIFAR10

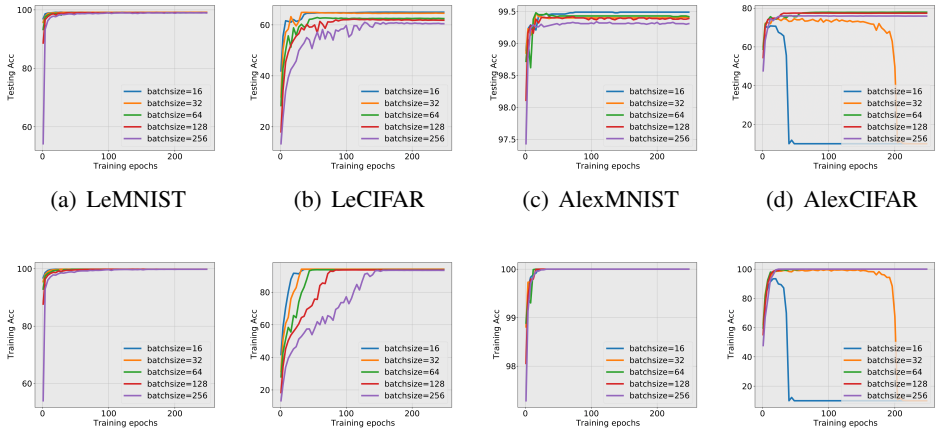
batchsize	spectral criterion $C = 0.4$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
16	24	61.37%	8	61.62%	BT	HT	64.99%
32	60	64.78%	10	57.94%	BT	HT	64.57%
64	-		28	59.19%	LT	BT	62.49%
128	-		60	61.38%	LT	BT	61.83%
256	-		84	54.23%	LT	BT	60.49%

The combination MiniAlexNet+MNIST

batchsize	spectral criterion $C = 0.4$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
16	4	99.23%	-		BT	LT	99.49%
32	20	99.42%	-		BT	LT	99.41%
64	-		-		LT	LT	99.42%
128	-		-		LT	LT	99.39%
256	-		-		LT	LT	99.31%

The combination MiniAlexNet+CIFAR10

batchsize	spectral criterion $C = 0.4$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
16	3	69.05%	9	72.02%	HT	RC	10%(explode)
32	4	72.17%	16	74.64%	HT	RC	10%(explode)
64	5	71.61%	28	76.35%	BT	BT	77.94%
128	10	74.14%	-		BT	LT	77.43%
256	24	75.70%	-		BT	LT	75.93%



(a) LeMNIST (b) LeCIFAR (c) AlexMNIST (d) AlexCIFAR

(e) LeMNIST (f) LeCIFAR (g) AlexMNIST (h) AlexCIFAR

Figure 13: Testing and Training Accuracy: We begin the line at epoch=1. Testing accuracy: (a)-(d); Training accuracy: (e)-(h). y-axis is the accuracy value, x-axis is the training epochs. Different line represents different batch sizes. (The notation "LeMNIST" means "LeNet+MNIST", same to "LeCIFAR", "AlexMNIST" and "AlexCIFAR").

5. Conclusion

The degree of difficulty of a classification problem has a great impact on the spectra of weight matrices. We study the phenomenon from three aspects: the SNR, the number of classes and the complexity of data features. We find that more difficult the classification is, the higher probability the heavy tails will emerge with. Further, in line with Martin and Mahoney (2021b), heavy tails could be regarded as a training information encoder and indicate some implicit regularization in NNs. Such implicit regularization in the weight matrices provides a new way of understanding the whole training process. Based on these findings, we derive a spectral criterion for early stopping. The procedure is capable of avoiding over-training when the data is of poor quality, and cutting off large training time when the classification problem is complex. From the encoded information, the spectral criterion can even provide an early stopped time when the training accuracy is still increasing.

Our study confirms that spectral analysis of weight matrices provides a new way for the understanding of Deep Learning. It also points to several unanswered questions to explore in the future. It seems particularly interesting to understand the reasons behind some of our empirical findings such as how SGD generates heavy tails with datasets from a difficult classification problem (even though the data may be of impeccable quality) but sticks to light tails with datasets from an easy or moderately difficult classification problem in deep neural networks.

Acknowledgements

The authors are particularly grateful to the numerous valuable comments received from the Editor and three referees. The final paper has much departed from its initial version, especially due to the revisions required by these very helpful comments.

References

Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, 2014.

Jungang Ge, Ying-Chang Liang, Zhidong Bai, and Guangming Pan. Large-dimensional random matrix theory and its applications in deep learning and wireless communications, 2021.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017. URL `http://arxiv.org/abs/1706.02677`.

Diego Granziol. Beyond random matrix theory for deep networks, 2020.

Mert Gurbuzbalaban, Umut Şimşekli, and Lingjiong Zhu. The heavy-tail phenomenon in sgd, 2021.

Liam Hodgkinson and Michael Mahoney. Multiplicative noise and heavy tails in stochastic optimization. In *International Conference on Machine Learning*, pages 4262–4274. PMLR, 2021.

Wenlong Ji, Yiping Lu, Yiliang Zhang, Zhun Deng, and Weijie J Su. How gradient descent separates data with neural collapse: A layer-peeled perspective. 2021.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016. URL `http://arxiv.org/abs/1609.04836`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Jan Kukacka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy, 2017.

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

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1EA-M-0Z`.

Charles H. Martin and Michael W. Mahoney. Post-mortem on a deep learning contest: a simpson's paradox and the complementary roles of scale metrics versus shape metrics. *CoRR*, abs/2106.00734, 2021a. URL https://arxiv.org/abs/2106.00734.

Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021b. URL http://jmlr.org/papers/v22/20-410.html.

Charles H Martin, Tongsu Serena Peng, and Michael W Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):1–13, 2021.

Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. *CoRR*, abs/1901.08244, 2019a. URL http://arxiv.org/abs/1901.08244.

Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size, 2019b.

Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra, 2020.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124005, dec 2019. doi: 10.1088/1742-5468/ab3bc3. URL https://doi.org/10.1088/1742-5468/ab3bc3.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond, 2017.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

Yaoqing Yang, Ryan Theisen, Liam Hodgkinson, Joseph E. Gonzalez, Kannan Ramchandran, Charles H. Martin, and Michael W. Mahoney. Evaluating natural language processing models with generalization metrics that do not need access to any training or testing data. *CoRR*, abs/2202.02842, 2022. URL https://arxiv.org/abs/2202.02842.

Jianfeng Yao, Shurong Zheng, and ZD Bai. *Sample covariance matrices and high-dimensional data analysis*. Cambridge University Press Cambridge, 2015.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W. Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 581–590, 2020. doi: 10.1109/BigData50022.2020.9378171.

Supplementary materials to 'Impact of classification difficulty on the weight matrices spectra in Deep Learning and application to early-stopping'

Xuran Meng, Jianfeng Yao*

Department of Statistics and Actuarial Science, The University of Hong Kong, Hong Kong SAR, China
and
School of Data Science, The Chinese University of Hong Kong (Shenzhen), China

* *To whom correspondence should be addressed: jeffyao@cuhk.edu.cn*

A. Useful results from random matrix theory

In this section, we review two results from random matrix theory (RMT) which are useful for our analysis. More complete information can be found in the literature, for example in the monograph Yao et al. (2015). As a model for the weight matrices in a DNN, consider an $n \times p$ random matrix $W = (w_{ij})$ where the entries $\{w_{ij}\}$ are i.i.d. complex random variables with mean zero and variance σ^2. Both dimensions p and $n = n(p)$ grow to infinity in such a way that $n/p \to c \in (0, \infty)$. The corresponding sample covariance matrix is $S_p = \frac{1}{p} WW^*$ and let $\lambda_1^{S_p} \geq \cdots \geq \lambda_n^{S_p}$ be its sorted eigenvalues. The ESD of S_p is

$$F^{S_p}(x) \triangleq \frac{1}{n} \sum_{j=1}^{n} I(\lambda_j \leq x), \quad x \in \mathbb{R},$$

where $I(\cdot)$ is the indicator function.

Theorem A.1 (Marchenko-Pastur law)**.** *Under the setting above, almost surely when $p \to \infty$, the ESD F^{S_p} converges to the Marchenko-Pastur law F_{c,σ^2} with parameter (c, σ^2) defined as follows: it has the density function*

$$p_{c,\sigma^2}(x) = \frac{1}{2\pi x c \sigma^2} \sqrt{(b-x)(x-a)}\, I(a < x < b), \quad x \in \mathbb{R}, \tag{A.1}$$

and if $c > 1$, an additional point mass of value $1 - 1/c$ at the origin. Here $a = \sigma^2(1 - \sqrt{c})^2$ and $b = \sigma^2(1 + \sqrt{c})^2$.

Theorem A.2 (Tracy-Widom Law)**.** *With the setting above and assume moreover that $\mathbb{E}w_{11}^4 < \infty$. Define*

$$\mu_{pn} = \frac{1}{p} \left\{ (p-1)^{\frac{1}{2}} + n^{\frac{1}{2}} \right\}^2$$

and

$$\sigma_{pn} = \frac{1}{p} \left\{ (p-1)^{\frac{1}{2}} + n^{\frac{1}{2}} \right\} \left\{ (p-1)^{-\frac{1}{2}} + n^{-\frac{1}{2}} \right\}^{\frac{1}{3}}.$$

Then as $p \to \infty$,

$$\lambda_1 \xrightarrow{a.s.} (1 + \sqrt{c})^2,$$

and

$$\frac{\lambda_1 - \mu_{pn}}{\sigma_{pn}} \xrightarrow{d} F_1,$$

where F_1 is the Tracy-Widom Law of order 1 whose distribution function is given by

$$F_1(s) = \exp\left\{\int_s^{+\infty} q(x) + (x-s)^2 q^2(x) dx\right\}, \quad s \in \mathbb{R},$$

where q solves the Painlevé II differential equation

$$q''(x) = xq(x) + 2q^3(x).$$

with boundary condition

$$q(s) \sim Ai(s),\ s \to +\infty.$$

Here $Ai(s)$ is the airy function.

B. Technical Proofs

In the following proofs, C denotes a generic constant that may change value from time to time.

B.1 Proof of Proposition 4.1

Let $h = h_M = (b-a)/M$ be the bin-size. By definition, $\hat{p}_M(x) = \frac{1}{nh}\sum_{i=1}^{n} I(X_i \in B(x))$,

$$\begin{aligned} s_n &= \int_a^b |\hat{p}_M(x) - p(x)|\, dx \\ &\leq \int_a^b |E\hat{p}_M(x) - p(x)|\, dx + \int_a^b |\hat{p}_M(x) - E\hat{p}_M(x)|\, dx. \end{aligned} \tag{B.1}$$

First term in (B.1): There exists $x^* \in B_1$ such that

$$P(X_i \in B_1) = C\int_a^{a+h} \frac{\sqrt{(b-x)(x-a)}}{x} dx = Ch\frac{\sqrt{(b-x^*)(x^*-a)}}{x^*} \leq C * \frac{1}{M} * \sqrt{\frac{1}{M}}.$$

Thus

$$P(X_i \in B_1) = O\left(\frac{1}{M^{\frac{3}{2}}}\right),$$

Similarly we also have $P(X_i \in B_M) = O\left(\frac{1}{M^{\frac{3}{2}}}\right)$. Then

$$\begin{aligned} &\int_{B_1 \cup B_M} |E\hat{p}_M(x) - p(x)|\, dx \\ &\leq \int_{B_1 \cup B_M} \{E\hat{p}_M(x) + p(x)\}\, dx \\ &= \int_{B_1 \cup B_M} \frac{1}{h} P(X_i \in B_1 \cup B_M)\, dx + P(X_i \in B_1 \cup B_M) \\ &= O\left(\frac{1}{M^{\frac{3}{2}}}\right). \end{aligned} \tag{B.2}$$

Consider a middle bin B_l with $l \in \{2, 3, ..., M-1\}$. There exists $x_l^* \in B_l$ such that for any $x \in B_l$, we have

$$E\hat{p}_M(x) = \frac{1}{h} P\left(X_i \in B\left(x\right)\right) = \frac{1}{h} \int_{a+lh}^{a+(l+1)h} p(u)du = p\left(x_l^*\right).$$

We can further find an x_l^{**} between x and x_l^*, such that

$$\begin{aligned} &\left|E\hat{p}_M\left(x\right) - p\left(x\right)\right| = \left|p\left(x_l^*\right) - p\left(x\right)\right| = \left|p'\left(x_l^{**}\right)\left(x_l^* - x\right)\right| \\ &\leq C * \frac{\left|x_l^{**} - \left(x_l^{**} - a\right)\left(b - x_l^{**}\right)\right|}{\sqrt{\left(x_l^{**} - a\right)\left(b - x_l^{**}\right)}\left(x_l^{**}\right)^2} * \left|x_l^* - x\right| \leq \frac{C}{\sqrt{lM}}. \end{aligned}$$

Then

$$\begin{aligned} &\int_{\cup_{l=1}^{M-1} B_l} \left|E\hat{p}_M\left(x\right) - p\left(x\right)\right| dx = \sum_{l=2}^{M-1} \int_{B_l} \left|E\hat{p}_M\left(x\right) - p\left(x\right)\right| dx \\ &\leq \sum_{l=2}^{M-1} \int_{B_l} \frac{1}{\sqrt{l}} \cdot \frac{C}{\sqrt{M}} dx = \sum_{l=2}^{M-1} \frac{1}{\sqrt{l}} \cdot \frac{C}{\sqrt{M}} \cdot \frac{1}{M} = O\left(\frac{1}{M}\right). \end{aligned}$$

Combining with (B.2), we have

$$\int_a^b \left|E\hat{p}_M\left(x\right) - p\left(x\right)\right| dx = O\left(\frac{1}{M}\right). \tag{B.3}$$

Second term in (B.1): We have

$$\begin{aligned} &P\left(\sup_x \left|\hat{p}_M\left(x\right) - E\hat{p}_M\left(x\right)\right| > \varepsilon\right) \\ &= P\left(M \cdot \max_{l=1,...,M} \tfrac{1}{n} \left|\sum_{i=1}^n I\left(X_i \in B_l\right) - nP\left(X_i \in B_l\right)\right| > \varepsilon\right) \\ &= P\left(\max_{l=1,...,M} \tfrac{1}{n} \left|\sum_{i=1}^n I\left(X_i \in B_l\right) - nP\left(X_i \in B_l\right)\right| > \tfrac{\varepsilon}{M}\right) \\ &\leq \sum_{l=1}^M P\left(\tfrac{1}{n} \left|\sum_{i=1}^n I\left(X_i \in B_l\right) - nP\left(X_i \in B_l\right)\right| > \tfrac{\varepsilon}{M}\right). \end{aligned}$$

Note that $P\left(X_i \in B_l\right) = O\left(1/M\right)$, by Bernstein inequality,

$$\sum_{l=1}^M P\left(\frac{1}{n}\left|\sum_{i=1}^n I\left(X_i \in B_l\right) - nP\left(X_i \in B_l\right)\right| > \frac{\varepsilon}{M}\right) \leq \sum_{l=1}^M e^{-C\frac{n^2\varepsilon^2}{M^2}\{\frac{n}{M} + \frac{n\varepsilon}{3M}\}^{-1}} \leq Me^{-C\frac{n\varepsilon}{M}}.$$

Let $\varepsilon = \sqrt{\frac{M \log n}{n}}$, we obtain

$$\sup_{x \in [a,b]} \left|\hat{p}_M\left(x\right) - E\hat{p}_M\left(x\right)\right| = O_p\left(\sqrt{\frac{M \log n}{n}}\right).$$

This implies that for any $\varepsilon_0 > 0$, there exists $R > 0$, such that

$$P\left(\sqrt{\frac{n}{M\log n}}\sup_{x\in[a,b]}|\hat{p}_M(x) - E\hat{p}_M(x)| > R\right) < \varepsilon_0.$$

On the other hand, for $\int_a^b |\hat{p}_M(x) - E\hat{p}_M(x)|\,dx$, there exists x_n satisfying

$$|\hat{p}_M(x_n) - E\hat{p}_M(x_n)|(b-a) \geq \int_a^b |\hat{p}_M(x) - E\hat{p}_M(x)|\,dx.$$

Combining these two estimates, we obtain

$$\begin{aligned}
\varepsilon_0 &> P\left(\sqrt{\frac{n}{M\log n}}\sup_{x\in[a,b]}|\hat{p}_M(x) - E\hat{p}_M(x)| > R\right) \\
&\geq P\left(\sqrt{\frac{n}{M\log n}}|\hat{p}_M(x_n) - E\hat{p}_M(x_n)|(b-a) > R(b-a)\right) \\
&\geq P\left(\sqrt{\frac{n}{M\log n}}\int_a^b |\hat{p}_M(x) - E\hat{p}_M(x)|\,dx > R(b-a)\right).
\end{aligned}$$

This is equivalent to the property:

$$\int_a^b |\hat{p}_M(x) - E\hat{p}_M(x)|\,dx = O_p\left(\sqrt{\frac{M\log n}{n}}\right).$$

Combining the estimates for the two terms in (B.1) completes the proof.

B.2 Proof of proposition 4.2

Recall the order statistics $X_{(1)} \leq \cdots \leq X_{(n)}$ of the sample. For $\varepsilon > 0$, there exists $R > 0$ such that

$$\begin{aligned}
&P\left(n^{\frac{2}{3}}\left(X_{(1)} - a\right) > R\right) = \prod_{i=1}^n P\left(X_i - a > R/n^{\frac{2}{3}}\right) \\
&= \left(1 - \int_a^{a+\frac{R}{n^{\frac{2}{3}}}} p(x)\,dx\right)^n \\
&\sim \left(1 - C\int_a^{a+\frac{R}{n^{\frac{2}{3}}}} \sqrt{x-a}dx\right)^n \\
&= \left(1 - CR^{\frac{3}{2}}/n\right)^n < \varepsilon,
\end{aligned}$$

for large enough n. Then $X_{(1)} - a = O_p\left(n^{-\frac{2}{3}}\right)$. Similarly, $b - X_{(n)} = O_p\left(n^{-\frac{2}{3}}\right)$. (The rate here is the same as in the Tracy-Widom Law).

With $p(x) = p_{c,\sigma^2}(x)$ and $\hat{p}(x) = p_{\hat{c},\hat{\sigma}^2}(x)$, we have by definition,

$$\hat{s}_n = \int_a^b |\hat{p}_M(x) - \hat{p}(x)|dx = \int_a^b |\hat{p}_M(x) - p(x) + \{p(x) - \hat{p}(x)\})|dx \leq s_n + \int_a^b |\hat{p}(x) - p(x)|dx.$$

Since the estimate for s_n is given in Proposition 4.1, we only need to estimate the last integral.

Let $C_0 = 2\pi\sigma^2 c$, $\hat{C}_0 = \frac{1}{2}\pi\left(\sqrt{X_{(n)}} - \sqrt{X_{(1)}}\right)^2$, we have

$$\int_a^b |\hat{p}(x) - p(x)|\, dx = \int_a^b \left| \frac{\sqrt{(b-x)(x-a)}}{C_0 x} - \frac{\sqrt{(X_{(n)}-x)(x-X_{(1)})}}{\hat{C}_0 x} I\left(\left[X_{(1)}, X_{(n)}\right]\right) \right| dx$$

$$\leq \int_a^b \left|\frac{1}{\hat{C}_0}\right| \cdot \left| \frac{\sqrt{(b-x)(x-a)}}{x} - \frac{\sqrt{(X_{(n)}-x)(x-X_{(1)})}}{x} I\left(\left[X_{(1)}, X_{(n)}\right]\right) \right| dx + \left|\frac{1}{C_0} - \frac{1}{\hat{C}_0}\right|$$

$$\triangleq P_1 + P_2.$$

Note that $\hat{C}_0 \xrightarrow{p} C_0 > 0$, by continuous mapping, $\frac{1}{\hat{C}_0} \xrightarrow{p} \frac{1}{C_0} > 0$, thus $\frac{1}{\hat{C}_0} = O_p(1)$. Then we have

$$\begin{aligned} P_2 = \left|\frac{1}{C_0} - \frac{1}{\hat{C}_0}\right| &= \frac{\pi\left(\left(\sqrt{b} - \sqrt{a}\right)^2 - \left(\sqrt{X_{(n)}} - \sqrt{X_{(1)}}\right)^2\right)}{2C_0\hat{C}_0} \\ &\leq O_p(1) \cdot O_p\left(\sqrt{b} - \sqrt{X_{(n)}} + \sqrt{X_{(1)}} - \sqrt{a}\right) = O_p\left(n^{-\frac{1}{3}}\right). \end{aligned}$$

For P_1, we have

$$P_1 = O_p(1) \int_a^b \left| \sqrt{(b-x)(x-a)} - \sqrt{(X_{(n)}-x)(x-X_{(1)})} I\left(\left[X_{(1)}, X_{(n)}\right]\right) \right| dx$$

Because $n^{2/3}(X_{(1)} - a) = O_p(1)$ and $n^{2/3}(X_{(n)} - b) = O_p(1)$, by continuous mapping, we have $n^{2/3}P_1 = O_p(1)$. Finally

$$\int_a^b |\hat{p}(x) - p(x)| dx \leq P_1 + P_2 = O_p\left(n^{-\frac{1}{3}}\right),$$

and the proof of the proposition is complete.

C. Algorithms

In this section, we detail algorithms for detection of spikes and estimate deviations from the MP Law.

Algorithm 1 below gives an automatic method of detecting spikes using a tuning parameter α. When the gap between spikes and bulk is larger than α times the average difference level, the gap will be detected by the algorithm. HS and TS represent the number of spikes larger or smaller than the value of bulk, respectively. The method is based on the principle that the gap between the spikes and the bulk is much larger than the average differences between subsequent eigenvalues.

After detecting the spikes with Algorithm 1, the deviation measurement between ESDs in weight matrices and standard MP Law is given by Algorithm 2 below. As described in Section 2.2.1, this algorithm is also used to distinguish the spectrum types BT and LT. Finally, Algorithm 2 calculates the spectral criterion value $\hat{s}_n$ introduced Section 4.

Algorithm 1 Auto-detection of spikes

Require: Eigenvalues $\{\lambda_i\}_{i=1}^{N}$, $\alpha \leftarrow 7$.
  Sort λ_i with descending order, $\lambda_i > \lambda_{i+1}$
  for $i = 1; i < N; i++$ **do**
    $\beta_i \leftarrow \lambda_i - \lambda_{i+1}$
  end for
  $\beta = \sum_{i=1}^{N-1} \beta_i/(N-1)$
  $r \leftarrow \alpha \cdot \beta$
  for $i = 1; i < N/2; i++$ **do**
    if $\beta_i > r$ **then**
      HS $\leftarrow i$
    end if
  end for
  for $i = N-1; i > N/2; i--$ **do**
    if $\beta_i > r$ **then**
      TS $\leftarrow N - i$
    end if
  end for
  return HS,TS

Algorithm 2 Get deviation measurement $\hat{s}_n$

Require: Eigenvalues $\{\lambda_i\}_{i=1}^{N}$, $\alpha \leftarrow 7$.
  Sort λ_i with descending order, $\lambda_i > \lambda_{i+1}$
  HS,TS $\leftarrow$ Algorithm 1($\{\lambda_i\}_{i=1}^{N}$, α)
  $n \leftarrow N-$HS$-$TS
  for $i = 1; i \leq n; i++$ **do**
    $\gamma_i \leftarrow \lambda_{i+\text{HS}}$ ▷ Get eigenvalues lying in the bulk
  end for
  $M \leftarrow 2\lfloor n^{\frac{1}{3}} \rfloor$ ▷ The number of Bins
  $H \leftarrow \lfloor n/M \rfloor$
  $f(x) \leftarrow \frac{2\sqrt{(\gamma_1 - x)(x - \gamma_n)}}{\pi(\sqrt{\gamma_1} - \sqrt{\gamma_n})^2 x}$
  $\hat{s}_n = 0$
  for $i = 1; i < M; i++$ **do**
    $a, b \leftarrow (i-1)H + 1, iH + 1$
    $L \leftarrow (b-a)/n/(\gamma_a - \gamma_b)$
    $s \leftarrow \int_{\gamma_b}^{\gamma_a} |f(x) - L| dx$
    $\hat{s}_n \leftarrow \hat{s}_n + s$
  end for
  $a, b \leftarrow (M-1)H + 1, n$
  $L \leftarrow (b-a)/n/(\gamma_a - \gamma_b)$
  $s \leftarrow \int_{\gamma_b}^{\gamma_a} |f(x) - L| dx$
  $\hat{s}_n \leftarrow \hat{s}_n + s$
  return $\hat{s}_n$

D. Results for the spectral criterion with $C = 0.6$

In this section, we report additional experimental results of the spectral criterion with critical constant $C = 0.6$ (results in the main text all use the value of $C = 0.4$). A higher value of C makes the criterion more conservative and the additional results with $C = 0.6$ help check the robustness of the spectral criterion.

A higher value of C thus implies a longer time of training. Detailed results are reported in Tables 9 and 10, for the experiments on the synthetic Gaussian data and the real data sets MNIST and CIFAR10, respectively. They parallel the results reported in Tables 7 and 8 of the paper using $C = 0.4$. To a very large extent, the new results confirm our conclusions in the previous tables. Again the spectral criterion detects the problematic issues in the numeric experiments and suggests early stopping even when the training accuracy is increasing. In the real data experiments, the spectral criterion predicts the training explosion quite accurately. Combined all the results with $C = 0.4$ and $C = 0.6$, we recommend the use of the spectral criterion with a critical constant C in the range of [0.4,0.6].

Table 9: Early stopping results in numeric experiments with $C = 0.6$: stopping epochs selected by spectral criterion in different layers' weight matrices and their testing accuracy (Test Acc). The symbol "-" means no early stopping epoch is found by the spectral criterion.

The combination NN1+$\mathcal{D}_1$

Typical TP	spectral criterion $C = 0.6$				Final Epoch 248		
	epoch(FC2)	Test Acc	epoch(FC3)	Test Acc	FC1	FC2	Test Acc
0.15	8	24.58%	16	20.44%	HT	HT	20.17%
0.2	8	31.50%	16	25.83%	HT	HT	27.03%
0.3	8	49.09%	12	45.48%	HT	HT	44.80%
0.6	9	87.96%	-		BT	BT	88.30%
0.9	-		-		LT	LT	99.13%

The combination NN1+$\mathcal{D}_2$

Typical TP	spectral criterion $C = 0.6$				Final Epoch 248		
	epoch(FC2)	Test Acc	epoch(FC3)	Test Acc	FC1	FC2	Test Acc
0.24	9	14.69%	-		HT	HT	13.08%
1.2	8	39.31%	16	32.11%	HT	HT	32.98%
2.4	8	76.75%	20	74.29%	HT	HT	75.92%
3.2	10	91.94%	-		HT	LT	92.64%
4.8	-		-		LT	LT	99.73%

The combination NN2+$\mathcal{D}_1$

Typical TP	spectral criterion $C = 0.6$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
0.02	-		-		HT	BT	16.02%
0.04	-		-		HT	BT	25.38%
0.07	-		-		HT	BT	50.12%
0.13	-		-		BT	LT	87.50%
0.2	-		-		LT	LT	99.14%

The combination NN2+$\mathcal{D}_2$

Typical TP	spectral criterion $C = 0.6$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
0.24	12	13.48%	7	13.19%	HT	HT	13.44%
1.2	24	35.80%	5	34.63%	HT	HT	36.31%
2.4	6	73.80%	36	74.86%	BT	BT	75.12%
3.2	-		-		LT	LT	91.20%
4.8	-		-		LT	LT	99.59%

Table 10: Early stopping results in real data experiments with $C = 0.6$.

The combination LeNet+MNIST

batchsize	spectral criterion $C = 0.6$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
16	-		32	99.19%	LT	BT	99.17%
32	-		-		LT	BT	99.17%
64	-		-		LT	BT	98.98%
128	-		-		LT	BT	99.03%
256	-		-		LT	LT	98.96%

The combination LeNet+CIFAR10

batchsize	spectral criterion $C = 0.6$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
16	28	61.66%	8	61.62%	BT	HT	64.99%
32	-		20	61.06%	BT	HT	64.57%
64	-		32	60.27%	LT	BT	62.49%
128	-		60	61.38%	LT	BT	61.83%
256	-		92	58.33%	LT	BT	60.49%

The combination MiniAlexNet+MNIST

batchsize	spectral criterion $C = 0.6$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
16	5	98.64%	-		BT	LT	99.49%
32	-		-		BT	LT	99.41%
64	-		-		LT	LT	99.42%
128	-		-		LT	LT	99.39%
256	-		-		LT	LT	99.31%

The combination MiniAlexNet+CIFAR10

batchsize	spectral criterion $C = 0.6$				Final Epoch 248		
	epoch(FC1)	Test Acc	epoch(FC2)	Test Acc	FC1	FC2	Test Acc
16	4	71.01%	36(RC)	55.6%	HT	RC	10%(explode)
32	4	72.17%	196(RC)	62.84%	HT	RC	10%(explode)
64	6	73.03%	-		BT	BT	77.94%
128	12	74.31%	-		BT	LT	77.43%
256	28	75.87%	-		BT	LT	75.93%