Deep frequency principle towards understanding why deeper learning is faster

Zhi-Qin John Xu\textsuperscript{a} and Hanxu Zhou\textsuperscript{a}

\textsuperscript{a}School of Mathematical Sciences, MOE-LSC and Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai, 200240, P.R. China

July 29, 2020

Abstract

Understanding the effect of depth in deep learning is a critical problem. In this work, we utilize the Fourier analysis to empirically provide a promising mechanism to understand why deeper learning is faster. To this end, we separate a deep neural network into two parts, one is a pre-condition component and the other is a learning component, in which the output of the pre-condition one is the input of the learning one. Based on experiments of deep networks and real dataset, we propose a deep frequency principle, that is, the effective target function for a deeper hidden layer has a bias towards a function with more low frequency during the training. Therefore, the learning component effectively learns a lower frequency function if the pre-condition component has more layers. Due to the well-studied frequency principle, i.e., deep neural networks learn lower frequency functions faster, the deep frequency principle provides a reasonable explanation to why deeper learning is faster. We believe these empirical studies would be valuable for future theoretical studies of the effect of depth in deep learning.

1 Introduction

Deep neural networks have achieved tremendous success in many applications, such as computer vision, speech recognition, speech translation, and natural language processing etc. The depth in neural networks plays an important role in the applications. Understanding the effect of depth is a central problem to reveal the “black box” of deep learning. For example, empirical studies show that a deeper network can learn faster and generalize better in both real data and synthetic data \cite{11,1}. Different network structures have different computation costs in each training epoch. In this work, we define that \textit{the learning of a deep neural network is faster if the loss of the deep neural network decreases to a designated error with fewer training epochs.} For example, as shown in Fig. 1(a), when learning data sampled from a target function $\cos(3x) + \cos(5x)$, a deep neural network with more hidden layers achieved the designated training loss with fewer training epochs. Although empirical studies suggest deeper neural network may learn faster, there is few understanding of the mechanism.

* xuzhiqin@sjtu.edu.cn
In this work, we would empirically explore an underlying mechanism that may explain why deeper neural network can learn faster from the perspective of Fourier analysis. We start from a universal phenomenon of frequency principle [30, 22, 31, 17, 29], that is, deep neural networks often fit target functions from low to high frequencies during the training. Recent works shows that F-Principle may provide an understanding to the success and failure of deep learning [31, 34, 29, 19]. We use an ideal example to illustrate the frequency principle, i.e., using a deep neural network to fit different target functions. As the frequency of the target function decreases, the deep neural network achieves a designated error with fewer training epochs, which is similar to the phenomenon when using a deeper network to learn a fixed target function.

Inspired by the above analysis, we propose a mechanism to understand why a deeper network, $f_\theta(x)$, faster learns a set of training data, $S = \{(x_i, y_i)\}_{i=1}^n$ sampled from the same target function $f^*(x)$, illustrated as follows. We separate a deep neural network into two parts, as shown in Fig. 3, one is a pre-condition component and the other is a learning component, in which the output of the pre-condition one, denoted as $f^{[l]}_\theta(x)$ (first $l$ layers are classified as the pre-condition component), is the input of the learning one. For the learning component, the effective training data at each training epoch is $S^{[l]} = \{(f^{[l]}_\theta(x_i), y_i)\}_{i=1}^n$. We then perform experiments based on the variants of Resnet18 structure [11] and CIFAR10 dataset. We fix the learning component (fully-connected layers). When increasing the number of the pre-condition layer (convolution layers), we find that $S^{[l]}$ contains more low frequency and less high frequency by frequency principle, the learning of a lower frequency function is faster, therefore, the learning component is faster to learn $S^{[l]}$ when the pre-condition component has more layers. The analysis among different network structures is often much more difficult than the analysis of one single structure. For providing hints for future theoretical study, we study a fixed fully-connected deep neural network by classifying different number of layers into the pre-condition component, i.e., varying $l$. As $l$ increases, we similarly find that $S^{[l]}$ contains more low frequency and less high frequency during the training. Therefore, we propose the following principle:

Deep frequency principle: The effective target function for a deeper hidden layer has a bias
towards a function with more low frequency during the training.

With the previous study on frequency principle, the deep frequency principle shows a promising mechanism for understanding why a deeper network learns faster.

1.1 Related work

From the perspective of approximation, the expressive power of a deep neural network increases with the depth [26, 9, 8]. However, the approximation theory renders no implication on the optimization of deep neural networks.

With residual connection, [11] successfully train very deep networks and find that deeper networks can achieve better generalization error. In addition, [11] also show that the training of deeper network is faster. [1] show that the acceleration effect of depth also exists in deep linear neural network and provide a viewpoint for understanding the effect of depth, that is, increasing depth can be seen as an acceleration procedure that combines momentum with adaptive learning rates. There are also many works studying the effect of depth for deep linear networks [24, 15, 10, 25].

Various studies suggest that the function learned by the deep neural networks increases its complexity as the training goes [3, 27, 20, 14, 33]. This increasing complexity is also found in deep linear network [10]. The high-dimensional experiments in [31] show that the low-frequency part is converged first, i.e., frequency principle. Therefore, the ratio of the low-frequency component of the deep neural network output experiences a increasing stage at the beginning (due to the convergence of low-frequency part), followed by a decreasing stage (due to the convergence of high-frequency part). As more high-frequency involved, the complexity of the deep neural network output increases. Therefore, the ratio of the low-frequency component used in this paper validates the complexity increasing during the training, which is consistent with other studies.

Frequency principle is examined in extensive datasets and deep neural networks [30, 22, 31]. Theoretical studies subsequently shows that frequency principle holds in general setting with infinite samples [17] and in the regime of wide neural networks (Neural Tangent Kernel (NTK) regime [12]) with finite samples [34] or infinite samples [6, 33, 23]. [7] shows that the integral equation would naturally leads to the frequency principle. With the theoretical understanding, the frequency principle inspires the design of deep neural networks to fast learn a function with high frequency [16, 28, 13, 5, 4].

2 Preliminary

2.1 Low Frequency Ratio (LFR)

To compare two 1-d functions in the frequency domain, we can display their spectrum. However, this does not apply for high-dimensional function because the computation cost of high-dimensional Fourier transform suffers from the curse of dimensionality. To overcome this, we use low-frequency filter to derive a low-frequency component of the interested function and then use a Low Frequency Ration (LFR) to characterize the ratio of power of the low-frequency component over the whole spectrum.

The LFR is defined as follows. We first split the frequency domain into two parts, i.e., a low-frequency part with frequency $|k| \leq k_0$ and a high-frequency part with $|k| > k_0$, where $|\cdot|$ is the length of a vector. Consider a dataset $\{(x_i, y_i)\}_{i=1}^{n}$, $x_i \in \mathbb{R}^d$, and $y_i \in \mathbb{R}^{d_o}$. For example, $d = 784$. 

and $d_o = 10$ for MNIST and $d = 3072$ and $d_o = 10$ for CIFAR10. The LFR is defined as

$$LFR = \frac{\sum_k \mathbb{1}_{|k| \leq k_0} |\hat{y}(k)|^2}{\sum_k |\hat{\theta}(k)|^2},$$

(1)

where $\hat{\cdot}$ indicates Fourier transform, $\mathbb{1}_{|k| \leq k_0}$ is an indicator function, i.e.,

$$\mathbb{1}_{|k| \leq k_0} = \begin{cases} 1, & |k| \leq k_0, \\ 0, & |k| > k_0. \end{cases}$$

However, because it is almost impossible to compute above quantities numerically due to high computational cost of high-dimensional Fourier transform, similarly as in [31], we alternatively use the Fourier transform of a Gaussian function $\hat{G}_\delta(k)$, where $\delta$ is the variance of the Gaussian function $G$, to approximate $\mathbb{1}_{|k| > k_0}$. Note that $1/\delta$ can be interpreted as the variance of $\hat{G}$. The approximation is reasonable due to the following two reasons. First, the Fourier transform of a Gaussian is still a Gaussian, i.e., $\hat{G}_\delta(k)$ decays exponentially as $|k|$ increases, therefore, it can approximate $\mathbb{1}_{|k| \leq k_0}$ by $\hat{G}_\delta(k)$ with a proper $\delta(k_0)$. Second, the computation of LFR contains the multiplication of Fourier transforms in the frequency domain, which is equivalent to the Fourier transform of a convolution in the spatial domain. We can equivalently perform the computation in the spatial domain so as to avoid the almost impossible high-dimensional Fourier transform. The low frequency part can be derived by

$$y_i^{\text{low}, \delta(k_0)} \triangleq (y * G^\delta(k_0))_i,$$

(2)

where $\ast$ indicates convolution operator. Then, we can compute the LFR by

$$LFR(k_0) = \frac{\sum_i |y_i^{\text{low}, \delta(k_0)}|^2}{\sum_i |y_i|^2},$$

(3)

The low frequency part can be derived on the discrete data points by

$$y_i^{\text{low}, \delta} = \frac{1}{C_i} \sum_{j=0}^{n-1} y_j G^\delta(x_i - x_j),$$

(4)

where $C_i = \sum_{j=0}^{n-1} G^\delta(x_i - x_j)$ is a normalization factor and

$$G^\delta(x_i - x_j) = \exp \left(-|x_i - x_j|^2/2\delta \right).$$

(5)

where $1/\delta$ is the variance of $\hat{G}$. $1/\delta$ can be interpreted as the frequency width outside which is filtered out by the convolution.

Note that ranges of different dimensions in the input are different, which would result in that for a same $\delta$, different dimensions keeps different frequency ranges when convolving with the Gaussian function. Therefore, we normalized each dimension by its maximum amplitude, thus, each dimension lies in $[-1, 1]$. 
2.2 Ratio Distribution Function (RDF)

LFR($k_0$) characterizes the power ratio of frequencies within a sphere of radius $k_0$. To characterize each frequency in the radius direction, similarly to probability, we define the ratio density function (RDF) as

$$\text{RDF}(k_0) = \frac{\partial \text{LFR}(k_0)}{\partial k_0}. \quad (6)$$

In practical computation, we use $1/\delta$ for $k_0$ and use the linear slope between two consecutive points for the derivative. For illustration, we show the LFR and RDF for $\sin(k\pi x)$ in Fig. 2. As shown in Fig. 2(a), the LFR of low-frequency function faster approaches one when the filter width in the frequency domain is small, i.e., small $1/\delta$. The RDF in Fig. 2(b) shows that as $k$ in the target function increases, the peak of RDF moves towards wider filter width. It is more intuitive that the RDF effectively reflects where the power of the function concentrates in the frequency domain. In the following, we will use RDF to study the frequency distribution of effective target functions for hidden layers.

![LFR and RDF for sin(kπx) vs. 1/δ](image)

Figure 2: LFR and RDF for $\sin(k\pi x)$ vs. $1/\delta$. Note that we normalize RDF in (b) by the maximal value of each curve for visualization.

2.3 General deep neural network

We adopt the notation in [21]. An $L$-layer neural network is denoted by

$$f_\theta(x) = W^{[L-1]}\sigma \circ (W^{[L-2]}\sigma \circ \cdots (W^{[1]}\sigma \circ (W^{[0]}x + b^{[0]}) + b^{[1]}) \cdots) + b^{[L-2]} + b^{[L-1]}, \quad (7)$$

where $W^{[l]} \in \mathbb{R}^{m_{l+1} \times m_l}$, $b^{[l]} = \mathbb{R}^{m_{l+1}}$, $m_0 = d_{in} = d$, $m_L = d_o$, $\sigma$ is a scalar function and “$\circ$” means entry-wise operation. We denote the set of parameters by

$$\theta = (W^{[0]}, W^{[1]}, \ldots, W^{[L-1]}, b^{[0]}, b^{[1]}, \ldots, b^{[L-1]}),$$
and an entry of $W^{[l]}$ by $W_{ij}^{[l]}$. This can also be defined recursively.

\begin{align}
  f^{[0]}_\theta(x) &= x, \\
  f^{[l]}_\theta(x) &= \sigma \circ (W^{[l-1]} f^{[l-1]}_\theta(x) + b^{[l-1]}) \quad 1 \leq l \leq L - 1, \\
  f_\theta(x) &= f^{[L]}_\theta(x) = W^{[L-1]} f^{[L-1]}_\theta(x) + b^{[L-1]}.
\end{align}

For simplicity, we also denote

\begin{equation}
  f^{[-l]}_\theta(x) = f^{[L-l+1]}_\theta(x).
\end{equation}

For example, the output layer is layer “−1”, i.e., $f^{[-1]}_\theta(x)$ for a given input $x$, and the last hidden layer is layer “−2”, i.e., $f^{[-2]}_\theta(x)$ for a given input $x$, illustrated in Fig. 3.

The effective target function for the learning component, consisting from layer “$l$” to the output layer, is

\begin{equation}
  S^{[l-1]} = \{(f^{[-1]}_\theta(x_i), y_i)\}_{i=1}^n.
\end{equation}

![Diagram](image)

Figure 3: General deep neural network

### 2.4 Training details

We list training details for experiments as follows.

For the experiments of the variants of Resnet18 on CIFAR10, the network structures are shown in Fig. 4. The output layer is equipped with softmax and the network is trained by Adam optimizer.
with cross-entropy loss and batch size 256. The learning rate is changed as the training proceeds, that is, $10^{-3}$ for epoch 1-40, $10^{-4}$ for epoch 41-60, and $10^{-5}$ for epoch 61-80. We use 40000 samples of CIFAR10 as training set and 10000 examples as the validation set. The training accuracy and the validation accuracy are shown in Fig. 5. The RDF of the effective target function of the last hidden layer for each variant is shown in 6.

For experiment of fully-connected network on MNIST, we choose the activation function of tanh and size 784 – 500 – 500 – 500 – 500 – 500 – 10. The output layer of the network does not equip any activation function. The network is trained by Adam optimizer with mean squared loss, batch size 256 and learning rate $10^{-5}$. The training is stopped when the loss is smaller than $10^{-2}$. We use 30000 samples of the MNIST as training set. The RDF of the effective target functions of different hidden layers are shown in Fig. 7.

Figure 4: Variants of Resnet18.
3 Results

Based on the experiments of very deep networks and real datasets, we would show a deep frequency principle, a promising mechanism, to understand why deeper neural networks learn faster, that is, the effective target function for a deeper hidden layer has a bias towards a function with more low frequency during the training. To derive the effective function, we decompose the target function into a pre-condition component, consisting of layers before the considered hidden layer, and a learning component, consisting from the considered hidden layer to the output layer, as shown in Fig. 3. As the hidden layer gets deeper, the learning component effectively learns a lower frequency function. Due to the frequency principle, i.e., deep neural networks learn low frequency faster, a deeper neural network can learn the target function faster. The key to validate deep frequency principle is to show the low frequency ratio distribution of the effective target function for each considered hidden layer.

First, we study a practical and common situation, that is, networks with more hidden layers learn faster. The further study on different networks may be much more difficult than the study of a single deep neural network, therefore, we then examine the deep frequency principle on a fixed deep neural network but consider the effective target function for different hidden layers.

3.1 Deep frequency principle on variants of Resnet18

In this subsection, we would utilize variants of Resnet18 and CIFAR10 dataset to validate deep frequency principle. The structures of four variants are illustrated as follows. As shown in Fig. 4, all structures have several convolution parts, followed by two same fully-connected layers. Compared with Resnet18-\(i\), Resnet18-(\(i+1\)) drops out a convolution part and keep other parts the same. The training details can be found in Section 2.4.

As shown in Fig. 5, a deeper net attains a fixed training accuracy with fewer training epochs and achieves a better generalization after training.

![Figure 5: Training accuracy and validation accuracy vs. epoch for variants of Resnet18.](image)

From the layer “-2” to the final output, it can be regarded as a two-layer neural network, which
is widely studied. Next, we examine the RDF for layer "-2". The effective target function is

\[ S^{[-3]} = \left\{ \left( f^{[-3]}_\theta(x_i), y_i \right) \right\}_{i=1}^n. \]  

(13)

As shown in Fig. 6(a), at initialization, the RDFs for deeper networks concentrate at higher frequency. However, as training proceeds, the concentration of RDFs of deeper networks move towards lower frequency. Therefore, for the two-layer neural network (NN) with a deeper pre-condition NN, the learning can be accelerated due to the fast convergence of low frequency in NN’s dynamics, i.e., frequency principle.

For the two-layer NN embedded as the learning component of the full network, the effective target function is \( S^{[-3]} \). As the pre-condition component has more layers, layer "-2" is a deeper hidden layer in the full network. Therefore, Fig. 6 validates that the effective target function for a deeper hidden layer has a bias towards a function with more low frequency during the training, i.e., deep frequency principle.

The comparison in the experiment in Fig. 6 crosses different networks, which would be difficult for future analysis. Alternatively, we can study how RDFs of \( S^{[-l]} \) of different \( l \) in a fixed deep network evolves during the training process. As expected, as \( l \) increases, \( S^{[-l]} \) would be dominated by more lower frequency during the training process.

![Figure 6: RDF of layer "-2" vs. \( \delta \) at different epochs for variants of Resnet18.](image)

3.2 RDF of different hidden layers in a fully-connected deep neural network

As analyzed above, examination of the deep frequency principle in a deep neural network would provide valuable insights for future theoretical study. A key problem is that different hidden layers often have different sizes, i.e., \( S^{[l]} \)'s have different input dimensions over different \( l \)'s. LFR is similar to a volume ratio, thus, it depends on the dimension. To control dimension variable, we consider a
fully-connected deep neural network with the same size for different hidden layers to learn MNIST dataset.

As shown in Fig. 7, at initialization, the peak of RDF for a deeper hidden layer also locates at a higher frequency. As the training goes, the peak of RDF of a deeper hidden layer moves towards low frequency faster. At the end of the training, the frequency of the RDF peak monotonically decreases as the hidden layer gets deeper. This indicates that the effective target function for a deeper hidden layer evolves faster towards a low-frequency function, that is, the deep frequency principle in a deep neural network.

![Graphs showing RDF at different epochs](image)

Figure 7: RDF of different hidden layers vs. 1/δ at different epochs for a fully-connected deep neural network when learning MNIST. The five different colored curves are for five hidden layers, respectively.

### 4 Discussion

In this work, we empirically show a deep frequency principle that provides a promising mechanism for understanding the effect of depth in deep learning. Specifically, based on the well-studied frequency principle, the deep frequency principle well explains why deeper learning can be faster. We believe the study of deep frequency principle would provide valuable insight for further theoretical understanding of deep neural networks. Next, we discuss the relation of this work to other studies and some implications.

#### 4.1 Kernel methods

Kernel methods, such as support vector machine and random feature model, are powerful at performing non-linearly separable data. An intuitive explanation is that when data are projected into a much higher dimensional space, they are closer to be linearly separable. From the perspective of Fourier analysis, we quantify this intuition through the low-frequency ratio. After projecting
data to higher dimensional space through the hidden layers, neural networks transform the high-dimensional target function into a lower frequency effective function. The deeper neural networks project data more than once into high dimensional space, which is equivalent to the combination of multiple kernel methods. In addition, neural networks not only learn the weight of kernels but also are able to learn the kernels, showing a much more capability compared with kernel methods.

4.2 Generalization

Frequency principle reveals a low-frequency bias of deep neural networks [30, 31], which provides qualitative understandings [32, 34] for the good generalization of neural networks in problems dominated by low frequencies, such as natural image classification tasks, and for the poor generalization in problems dominated by high frequencies, such as predicting parity function. Generalization in real world problems [11] is often better as the network goes deeper. How to characterize the better generalization of deeper network is also a critical problem in deep learning. This work, validating a deep frequency principle, may provide more understanding to this problem in future work. As the network goes deeper, the effective target function for the last hidden layer is more dominated by low frequency. This deep frequency principle phenomenon is widely observed, even in fitting high-frequency function, such as parity function in our experiments. This suggest that deeper network may have more bias towards low frequency. However, it is difficult to examine the frequency distribution of the learned function on the whole Fourier domain due to the high dimensionality of data. In addition, since the generalization increment of a deeper network is more subtle, we are exploring a more precise characterization of the frequency distribution of a high-dimensional function.

4.3 How deep is enough?

The effect of depth can be intuitively understood as a pre-condition that transforms the target function to a lower frequency function. Qualitatively, it requires more layers to fit a higher frequency function. However, the effect of depth can be saturated. For example, the effective target functions for very deep layers can be very similar in the Fourier domain (dominated by very low frequency components) when the layer number is large enough, as an example shown in Fig. 7(g, h). A too deep network would cause extra waste of computation cost. A further study of the deep frequency principle may also provide a guidance for design the depth of the network structure.

Taken together, the deep frequency principle proposed in this work may have fruitful implication for future study of deep learning. A detailed study of deep frequency principle may require analyze different dynamical regimes of neural networks. As an example, a recent work [18] draws a phase diagram for two-layer ReLU neural networks at infinite width limit by three regimes, linear, critical and condensed regimes. Such study could inspire the study of phase diagram of deep neural networks. The linear regime is well studied [12, 34, 35, 2], which may be a good starting point and shed lights on the study of other regimes.

5 Acknowledgement

This work is sponsored by National Key R&D Program of China (2019YFA0709503) (Z. X.), Shanghai Sailing Program (Z. X.).
References

[1] Sanjeev Arora, N Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In 35th International Conference on Machine Learning, 2018.

[2] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. arXiv preprint arXiv:1904.11955, 2019.

[3] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. International Conference on Machine Learning, 2017.

[4] Simon Biland, Vinicius C Azevedo, Byungsoo Kim, and Barbara Solenthaler. Frequency-aware reconstruction of fluid simulations with generative networks. arXiv preprint arXiv:1912.08776, 2019.

[5] Wei Cai, Xiaoguang Li, and Lizuo Liu. A phase shift deep neural network for high frequency approximation and wave problems. Accepted by SISC, arXiv:1909.11759, 2019.

[6] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. arXiv preprint arXiv:1912.01198, 2019.

[7] Weinan E, Chao Ma, and Lei Wu. Machine learning from a continuous viewpoint. arXiv preprint arXiv:1912.12777, 2019.

[8] Weinan E and Wang Qingcan. Exponential convergence of the deep neural network approximation for analytic functions. SCIENCE CHINA Mathematics, 61(10):1733, 2018.

[9] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In Conference on learning theory, pages 907–940, 2016.

[10] Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. The implicit bias of depth: How incremental learning drives generalization. In International Conference on Learning Representations, 2019.

[11] 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.

[12] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In Advances in neural information processing systems, pages 8571–8580, 2018.

[13] Ameya D Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics, 404:109136, 2020.
[14] Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and HaoFeng Zhang. Sgd on neural networks learns functions of increasing complexity. In Advances in Neural Information Processing Systems, pages 3491–3501, 2019.

[15] Kenji Kawaguchi, Jiaoyang Huang, and Leslie Pack Kaelbling. Effect of depth and width on local minima in deep learning. Neural computation, 31(7):1462–1498, 2019.

[16] Ziqi Liu, Wei Cai, and Zhi-Qin John Xu. Multi-scale deep neural network (mscalednn) for solving poisson-boltzmann equation in complex domains. arXiv preprint arXiv:2007.11207, 2020.

[17] Tao Luo, Zheng Ma, Zhi-Qin John Xu, and Yaoyu Zhang. Theory of the frequency principle for general deep neural networks. arXiv preprint arXiv:1906.09255, 2019.

[18] Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang. Phase diagram for two-layer relu neural networks at infinite-width limit. arXiv preprint arXiv:2007.07497, 2020.

[19] Chao Ma, Lei Wu, and Weinan E. The slow deterioration of the generalization error of the random feature model. In Proceedings of Machine Learning Research, 2020.

[20] Chris Mingard, Joar Skalse, Guillermo Valle-Pérez, David Martínez-Rubio, Vladimir Mikulik, and Ard A Louis. Neural networks are a priori biased towards boolean functions with low entropy. arXiv preprint arXiv:1909.11522, 2019.

[21] Beijing Academy of Artificial Intelligence. Suggested notation for machine learning. [https://github.com/Mayuyu/suggested-notation-for-machine-learning], 2020.

[22] Nasim Rahaman, Devansh Arpit, Aristide Baratin, Felix Draxler, Min Lin, Fred A Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of deep neural networks. International Conference on Machine Learning, 2019.

[23] Basri Ronen, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In Advances in Neural Information Processing Systems, pages 4763–4772, 2019.

[24] Andrew M Saxe, James L Mcclelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural network. In International Conference on Learning Representations. Citeseer, 2014.

[25] Yeonjong Shin. Effects of depth, width, and initialization: A convergence analysis of layer-wise training for deep linear neural networks. arXiv preprint arXiv:1910.05874, 2019.

[26] Matus Telgarsky. benefits of depth in neural networks. In Conference on Learning Theory, pages 1517–1539, 2016.

[27] Guillermo Valle-Perez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. arXiv preprint arXiv:1805.08522, 2018.

[28] Feng Wang, Alberto Eljarrat, Johannes Müller, Trond R Henninen, Rolf Erni, and Christoph T Koch. Multi-resolution convolutional neural networks for inverse problems. Scientific reports, 10(1):1–11, 2020.
[29] E Weinan, Chao Ma, and Lei Wu. Machine learning from a continuous viewpoint. arXiv preprint arXiv:1912.12777, 2019.

[30] Zhi-Qin J Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. International Conference on Neural Information Processing, pages 264–274, 2019.

[31] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. Accepted by Communications in Computational Physics, arXiv:1901.06523, 2019.

[32] Zhiqin John Xu. Understanding training and generalization in deep learning by fourier analysis. arXiv preprint arXiv:1808.04295, 2018.

[33] Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks. arXiv preprint arXiv:1907.10599, 2019.

[34] Yaoyu Zhang, Zhi-Qin John Xu, Tao Luo, and Zheng Ma. Explicitizing an implicit bias of the frequency principle in two-layer neural networks. arXiv preprint arXiv:1905.10264, 2019.

[35] Yaoyu Zhang, Zhi-Qin John Xu, Tao Luo, and Zheng Ma. A type of generalization error induced by initialization in deep neural networks. arXiv preprint arXiv:1905.07777, 2019.