统计计算与机器学习*Lecture 2 and 3: Generalization and frequency perspective

许志钦, xuzhiqin@sjtu.edu.cn

2020 年 3 月 20 日

Summary

This section focus on generalization of deep Learning and how to utilize frequency perspective to study generalization.

请不要上传到网络公开分享。手稿可能有很多错误，不严格的地方。

1 Fourier analysis

1.1 Definition

The conventional definition of Fourier transforms (FT) in signal processing is as follows.

- Continuous FT (CFT)

$$\mathcal{F}[g(x)](\xi)=\int_{-\infty}^{\infty}g(x)\mathrm{e}^{-2\pi\mathrm{i}\xi x}\,\mathrm{d}x,\qquad \mathcal{F}^{-1}[\hat{g}(\xi)](x)=\int_{-\infty}^{\infty}\hat{g}(\xi)\mathrm{e}^{2\pi\mathrm{i}\xi x}\,\mathrm{d}\xi \tag{1}$$

- Discrete-Time FT (DTFT) ($-\frac{1}{2\Delta}\le\xi\le\frac{1}{2\Delta}$, $x=j\Delta$, $j\in\mathbb{Z}$)

$$\mathcal{F}_{DTFT,\Delta}[g(x)](\xi)=\hat{g}_{DTFT}(\xi)=\sum_{j=-\infty}^{\infty}g(j\Delta)\mathrm{e}^{-2\pi\mathrm{i}\xi j\Delta} \tag{2}$$

$$\mathcal{F}^{-1}_{DTFT,\Delta}[\hat{g}(\xi)](j)=\Delta\cdot\int_{-\frac{1}{2\Delta}}^{\frac{1}{2\Delta}}\hat{g}_{DTFT}(\xi)\mathrm{e}^{2\pi\mathrm{i}\xi j\Delta}\,\mathrm{d}\xi \tag{3}$$

- Fourier Series (FS) ($-\frac{T}{2}\le x\le\frac{T}{2}$, $\xi=\frac{k}{T}$, $k\in\mathbb{Z}$)

$$\mathcal{F}_{FS,T}[g(x)](k)=c_k=\frac{1}{T}\cdot\int_{-\frac{T}{2}}^{\frac{T}{2}}g(x)\mathrm{e}^{-2\pi\mathrm{i}tk/T}\mathrm{d}x\qquad \mathcal{F}^{-1}_{FS,T}[c_k](x)=\sum_{k=-\infty}^{\infty}c_k\mathrm{e}^{2\pi\mathrm{i}xk/T} \tag{4}$$

The $\{c_k\}_{k\in\mathbb{Z}}$ above is called Fourier coefficients.

*上海交通大学2020春本科课程

- Discrete FT (DFT) $(j,k \in 0,1,\dots N-1)$

$$\mathcal{F}_{DFT}\left[\{a_j\}_{j=0}^{N-1}\right](k) = \sum_{j=0}^{N-1} a_j \mathrm{e}^{-2\pi \mathrm{i} kj/N} \qquad \mathcal{F}_{DFT}^{-1}\left[\{b_k\}_{k=0}^{N-1}\right](j) = \frac{1}{N}\sum_{k=0}^{N-1} b_k \mathrm{e}^{2\pi \mathrm{i} kj/N} \tag{5}$$

Note that if the sampling interval is finite-width, then there is a smallest frequency (discrete frequency); if the sampling is discrete, then, there is a maximum frequency (finite-interval frequency range). That is, band limite in one domain equivalent to discretize in another domain.

1.2 From Continuous FT to Discrete-Time FT (or Fourier Series)

$$\mathcal{F}_{DTFT,\Delta}[g(x)](\xi) = \sum_{j=-\infty}^{\infty} g(j\Delta)\mathrm{e}^{-2\pi \mathrm{i}\xi j\Delta} \tag{6}$$

$$= \sum_{j=-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{g}(\xi')\mathrm{e}^{2\pi \mathrm{i}\xi' j\Delta}\,\mathrm{d}\xi' \mathrm{e}^{-2\pi \mathrm{i}\xi j\Delta} \tag{7}$$

$$= \sum_{j=-\infty}^{\infty}\sum_{M=-\infty}^{\infty} \int_{\frac{M}{\Delta}-\frac{1}{2\Delta}}^{\frac{M}{\Delta}+\frac{1}{2\Delta}} \hat{g}(\xi')\mathrm{e}^{2\pi \mathrm{i}\xi' j\Delta}\,\mathrm{d}\xi' \mathrm{e}^{-2\pi \mathrm{i}\xi j\Delta} \tag{8}$$

$$= \sum_{j=-\infty}^{\infty}\sum_{M=-\infty}^{\infty} \int_{-\frac{1}{2\Delta}}^{\frac{1}{2\Delta}} \hat{g}(\xi'+\frac{M}{\Delta})\mathrm{e}^{2\pi \mathrm{i}\xi' j\Delta}\,\mathrm{d}\xi' \mathrm{e}^{-2\pi \mathrm{i}\xi j\Delta} \tag{9}$$

Define $\hat{h}(\xi') = \hat{g}(\xi' + \frac{M}{\Delta})$,

$$\mathcal{F}_{DTFT,\Delta}[g(x)](\xi) = \frac{1}{\Delta}\sum_{M=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\left(\Delta\int_{-\frac{1}{2\Delta}}^{\frac{1}{2\Delta}} \hat{h}(\xi')\mathrm{e}^{2\pi \mathrm{i}\xi' j\Delta}\,\mathrm{d}\xi'\right)\mathrm{e}^{-2\pi \mathrm{i}\xi j\Delta}. \tag{10}$$

By using inverse DTFT,

$$\mathcal{F}_{DTFT,\Delta}[g(x)](\xi) = \frac{1}{\Delta}\sum_{M=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} h(j\Delta)\mathrm{e}^{-2\pi \mathrm{i}\xi j\Delta}. \tag{11}$$

By using DTFT,

$$\mathcal{F}_{DTFT,\Delta}[g(x)](\xi) = \frac{1}{\Delta}\sum_{M=-\infty}^{\infty} \hat{h}(\xi) \tag{12}$$

$$= \frac{1}{\Delta}\sum_{M=-\infty}^{\infty} \hat{g}(\xi+\frac{M}{\Delta}). \tag{13}$$

Therefore, for any $\hat{g}(\xi)$ is a periodic function. There is not meaning to discuss frequency which is larger than $\frac{M}{2\Delta}$. In the discrete sampling, the signal of frequency $\xi \in [-\frac{1}{2\Delta}, \frac{1}{2\Delta}]$ is the summation of many frequency of the original singal. This is the aliasing phenomenon.

The Nyquist Sampling Theorem states that: A bandlimited continuous-time signal can be sampled and perfectly reconstructed from its samples if the waveform is sampled over twice as fast as it's highest frequency component.

Homework Prove the signal reconstructed from the inverse of DTFT in Eq. (3) is the same as the true function at $x_j = j\Delta$.

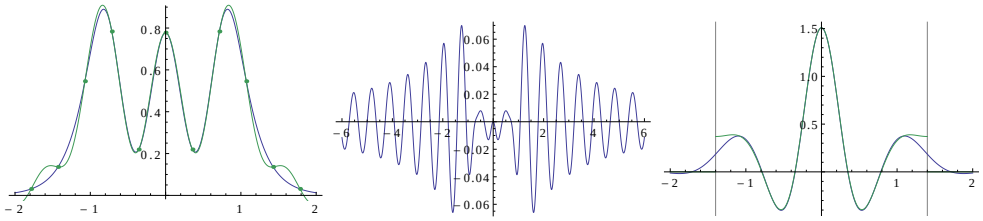
Idea: denoted the reconstructed signal by DTFT as g_{DTFT},

$$g_{DTFT}(x) - g(x) = \Delta \cdot \int_{-\frac{1}{2\Delta}}^{\frac{1}{2\Delta}} \hat{g}_{DTFT}(\xi) \mathrm{e}^{2\pi \mathrm{i}\xi j\Delta} \,\mathrm{d}\xi - \int_{-\infty}^{-\infty} \hat{g}(\xi) \mathrm{e}^{2\pi \mathrm{i}\xi j\Delta} \,\mathrm{d}\xi,$$

where

$$\Delta \cdot \hat{g}_{DTFT}(\xi) = \sum_{M=-\infty}^{\infty} \hat{g}(\xi + \frac{M}{\Delta}). \tag{14}$$

The following Fig. 1 shows an example.



(a) Time domain (b) Time domain difference (c) Frequency domain

图 1: CFT v.s. DTFT. Blue curve in (a)(c) is the original signal (CFT). Green curves are the sampled Frequency domain signal (c) and reconstructed continuous signal (a).

1.3 Decay rate

Consider the Fourier transform of the following $\delta(x)$ function,

$$\delta(x) = \begin{cases} \infty & x = 0 \\ 0 & x \neq 0 \end{cases} \quad \text{and} \quad \int \delta(x)\mathrm{d}x = 1. \tag{15}$$

Using Eq. (1),

$$\mathcal{F}[\delta(x)](\xi) = \int_{-\infty}^{\infty} \delta(x) \mathrm{e}^{-2\pi \mathrm{i}\xi x} \,\mathrm{d}x = 1. \tag{16}$$

The FT of $\delta(x)$ is a constant, which does not decay. Consider the Heaviside function

$$H(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}. \tag{17}$$

The derivative of $H(x)$ is $\delta(x)$. The FT is as follows

$$\mathcal{F}[H(x)](\xi) = \int_{-\infty}^{\infty} H(x) \mathrm{e}^{-2\pi \mathrm{i}\xi x} \,\mathrm{d}x, \tag{18}$$

and

$$H(x) = \int_{-\infty}^{\infty} \hat{H}(\xi) \mathrm{e}^{2\pi \mathrm{i}\xi x} \,\mathrm{d}\xi. \tag{19}$$

Take derivative of the above equation at both sides,

$$\delta(x)=\int_{-\infty}^{\infty}2\pi\mathrm{i}\xi\hat{H}(\xi)\mathrm{e}^{2\pi\mathrm{i}\xi x}\,\mathrm{d}\xi=\int_{-\infty}^{\infty}\hat{\delta}(\xi)\mathrm{e}^{2\pi\mathrm{i}\xi x}\,\mathrm{d}\xi. \tag{20}$$

Then, we have

$$\hat{H}(\xi)=\frac{1}{2\pi\mathrm{i}\xi}. \tag{21}$$

Therefore, for the Heaviside function, its Fourier component's amplitude decays as $1/|\xi|$. Similarly, we can prove that if a function is continuous up to α order, its its Fourier component's amplitude decays as $1/|\xi|^{\alpha+2}$. Note the integration of $H(x)$ is continuous, we denote $\alpha=-1$ for $H(x)$.

2 Introduction of F-Principle

Deep neural networks (DNNs) have achieved tremendous success in many applications, such as computer vision, speech recognition, speech translation, and natural language processing etc. However, DNN sometimes fails and causes critical issues in applications. For example, in the application of auto-drive, DNN can recognize a stop sign as a speed limit sign due to an invisible perturbation. Such a "black-box" system has permeated many aspects of daily life and important industries. It is as much urgent and important to provide a satisfactory interpretation for DNN as the understanding of nature.

A key to understand DNNs is to study the error of DNNs in learning problems. The error can be decomposed into three types as follows. Approximation error measures the distance between the target function and the best function in the hypothesis set. According to the universal approximation theorem [1], a sufficient wide DNN with at least one hidden layer can approximate any function to a desired precision. Note that the activation function cannot be polynomial. Generalization error measures the distance between the best function in the hypothesis set and the best function learned by the DNN based on given samples. Note that a large DNN is often able to represent the target function, i.e., the approximation is close to zero, and the best function in the hypothesis set is often unknown. The generalization error often refers to the distance between the target function and the function learned by the DNN based on given samples. Optimization error measures the distance between the best function learned by the DNN based on given samples and the function learned by the DNN based on the given algorithm and given samples.

A large DNN is often able to represent the target function, i.e., the approximation is close to zero. Although the optimization problem of deep learning is highly non-convex, emiprical studies found the gradient-descent-based methods often find the global minimum, i.e., optimization error is close to zero. However, the generalization error varies in different training algorithms and datasets. It remains an problem to study the generalization error of deep learning.

One interesting problem is that the generalization performance seems violate the traditional wisdom, that is, DNNs often generalizes well although the number parameters is much larger than the number of samples. Traditional statistical theory suggests that as the model complexity (e.g., the parameter number) increases, the model will finally overfit the training data, i.e., generalize badly. For example of Runge's phenomenon, a low-order polynomial function may not perfectly fit the noisy

training data well, but it may recover the true function well; however, a very high order polynomial function often fit the noisy training data well, but it osicalltes significantly, which often leads to a very bad generalization. In the case of DNN, although the DNN is capable of fitting the training with highly oscillated function, it often learns the data by a relative flat curve [5]. Such one dimensional case is similar to the high dimensional real problem [11].

This oscillation and flatness from low-dimensional examples inspires us to study the training process of DNNs in the Fourier domain. A Frequency Principle (F-Principle) as follows [6, 9, 3]:

DNNs often fit target functions from low to high frequencies during the training.

[6] and [3] independent found the F-Principle ("spectral bias" in [3]) with numerical simulations on synthetic data and very limited real data. [10] subsequently propose a theoretical analysis framework for one hidden layer neural network with 1-d input, which illustrates the key mechanism underlying the F-Principle—the activation function (including tanh and Relu) in the Fourier domain decays as frequency increases. examined the F-Principle through classification problems on benchmark datasets with cross-entropy loss with a projection method and solving a Poisson equation with a variation loss function. [7] also proposes that DNN can be adopted to accelerate the convergence of low frequencies for scientific computing problems, in which most of the conventional methods (e.g., Jacobi method) exhibit the opposite convergence behavior—faster convergence for higher frequencies. [9] systematically summarizes some works in [10, 7] and push further the empirical study of the F-Principle in more details. Especially, [9] proposes a Gaussian filtering method which can verify the F-Principle in high dimensional datasets for both regression and classification problems and utilize the F-Principle to understand both the success and failure of DNNs in different types of problems.

There have been much progress on the F-Principle since it is found recently. The study of the training process from the frequency perspective makes important progress in understanding the strength and weakness of DNN, such as generalization and converging speed etc., which may consist in "a reasonably complete picture about the main reasons behind the success of modern machine learning" ([4]).

3 Empirical study of F-Principle

3.1 one-dimensional experiments

To illustrate the phenomenon of F-Principle, we use 1-d synthetic data to show the evolution of relative training error at different frequencies during the training of DNN. we train a DNN to fit a 1-d target function $f(x) = \sin(x) + \sin(3x) + \sin(5x)$ of three frequency components. On $n = 201$ evenly spaced training samples, i.e., $\{x_i\}_{i=0}^{n-1}$ in $[-3.14, 3.14]$, the discrete Fourier transform (DFT) of $f(x)$ or the DNN output (denoted by $h(x)$) is computed by $\hat{f}_k = \frac{1}{n}\sum_{i=0}^{n-1} f(x_i)\mathrm{e}^{-I2\pi ik/n}$ and $\hat{h}_k = \frac{1}{n}\sum_{i=0}^{n-1} h(x_i)\mathrm{e}^{-I2\pi jk/n}$, where k is the frequency. As shown in Fig. 2(a), the target function has three important frequencies as we design (black dots at the inset in Fig. 2(a)). To examine the convergence behavior of different frequency components during the training with MSE, we compute the relative

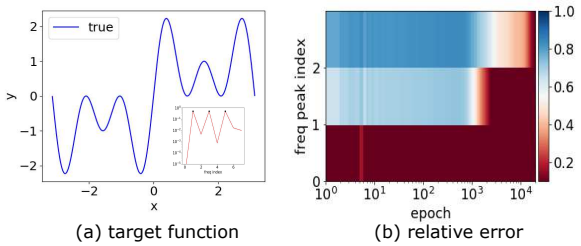


图 2: 1d input. (a) $f(x)$. Inset : $|\hat{f}(k)|$. (b) $\Delta_F(k)$ of three important frequencies (indicated by black dots in the inset of (a)) against different training epochs.

difference between the DNN output and the target function for the three important frequencies k's at each recording step, that is, $\Delta_F(k) = |\hat{h}_k - \hat{f}_k|/|\hat{f}_k|$, where $|\cdot|$ denotes the norm of a complex number. As shown in Fig. 2(b), the DNN converges the first frequency peak very fast, while converging the second frequency peak much slower, followed by the third frequency peak.

3.2 Frequency in high-dimensional classification problems

The concept of "frequency" is central to the understanding of F-Principle. In this paper, *the "frequency" means* ***response frequency*** NOT image (or input) frequency as explained in the following.

Image (or input) frequency (NOT used in the paper): Frequency of 2-d function $I : \mathbb{R}^2 \to \mathbb{R}$ representing the intensity of an image over pixels at different locations. This frequency corresponds to the rate of change of intensity *across neighbouring pixels*. For example, an image of constant intensity possesses only the zero frequency, i.e., the lowest frequency, while a sharp edge contributes to high frequencies of the image.

Response frequency (used in the paper): Frequency of a general Input-Output mapping f. For example, consider a simplified classification problem of partial MNIST data using only the data with label 0 and 1, $f(x_1, x_2, \cdots, x_{784}) : \mathbb{R}^{784} \to \{0, 1\}$ mapping 784-d space of pixel values to 1-d space, where x_j is the intensity of the j-th pixel. Denote the mapping's Fourier transform as $\hat{f}(k_1, k_2, \cdots, k_{784})$. The frequency in the coordinate k_j measures the rate of change of $f(x_1, x_2, \cdots, x_{784})$ *with respect to* x_j*, i.e., the intensity of the* j*-th pixel*. If f possesses significant high frequencies for large k_j, then a small change of x_j in the image might induce a large change of the output (e.g., adversarial example). For a dataset with multiple classes, we can similarly define frequency for each output dimension. For real data, the response frequency is rigorously defined via the standard nonuniform discrete Fourier transform (NUDFT) as follows.

In all our experiments, we consistently consider the response frequency defined for the mapping function g between inputs and outputs, say $\mathbb{R}^d \to \mathbb{R}$ and any $\boldsymbol{k} \in \mathbb{R}^d$ via the standard nonuniform discrete Fourier transform (NUDFT)

$$\hat{g}_{\boldsymbol{k}} = \frac{1}{n} \sum_{i=0}^{n-1} g(\boldsymbol{x}_i) \mathrm{e}^{-\mathrm{i}2\pi \boldsymbol{k} \cdot \boldsymbol{x}_i}, \tag{22}$$

which is a natural estimator of frequency composition of g. As $n \to \infty$, $\hat{g}_{\boldsymbol{k}} \to \int g(\boldsymbol{x}) \mathrm{e}^{-\mathrm{i}2\pi \boldsymbol{k}\cdot\boldsymbol{x}} \nu(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$, where $\nu(\boldsymbol{x})$ is the data distribution.

We restrict all the evaluation of Fourier transform in our experiments to NUDFT of $\{\boldsymbol{y}_i\}_{i=0}^{n-1}$ at $\{\boldsymbol{x}_i\}_{i=0}^{n-1}$ for the following practical reasons.

(i) The information of target function is only available at $\{\boldsymbol{x}_i\}_{i=0}^{n-1}$ for training.

(ii) It allows us to perform the convergence analysis. As $t \to \infty$, in general, $h(\boldsymbol{x}_i, t) \to \boldsymbol{y}_i$ for any i ($h(\boldsymbol{x}_i, t)$ is the DNN output), leading to $\hat{h}_{\boldsymbol{k}} \to \hat{y}_{\boldsymbol{k}}$ for any $\boldsymbol{k}$. Therefore, we can analyze the convergence at different $\boldsymbol{k}$ by evaluating $\Delta_F(\boldsymbol{k}) = |\hat{h}_{\boldsymbol{k}} - \hat{y}_{\boldsymbol{k}}|/|\hat{y}_{\boldsymbol{k}}|$ during the training. If we use a different set of data points for frequency evaluation of DNN output, then $\Delta_F(\boldsymbol{k})$ may not converge to 0 at the end of training.

(iii) $\hat{y}_{\boldsymbol{k}}$ faithfully reflects the frequency structure of training data $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=0}^{n-1}$. Intuitively, high frequencies of $\hat{y}_{\boldsymbol{k}}$ correspond to sharp changes of output for some nearby points in the training data. Then, by applying a Gaussian filter and evaluating still at $\{\boldsymbol{x}_i\}_{i=0}^{n-1}$, we obtain the low frequency part of training data with these sharp changes (high frequencies) well suppressed.

In practice, it is impossible to evaluate and compare the convergence of all $\boldsymbol{k} \in \mathbb{R}^d$ even with a proper cutoff frequency for a very large d of $O(10^2)$ (MNIST) or $O(10^3)$ (CIFAR10) due to curse of dimensionality. Therefore, we propose the projection approach, i.e., fixing $\boldsymbol{k}$ at a specific direction, and the filtering approach as detailed in Section 3 and 4, respectively.

3.3 Empirical study in high-dimensional classification problems

3.3.1 Examination method: Projection

For a dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=0}^{n-1}$ we consider one entry of 10-d output, denoted by $y_i \in \mathbb{R}$. The high dimensional discrete non-uniform Fourier transform of $\{(\boldsymbol{x}_i, y_i)\}_{i=0}^{n-1}$ is $\hat{y}_{\boldsymbol{k}} = \frac{1}{n}\sum_{i=0}^{n-1} y_i \exp\left(-I2\pi \boldsymbol{k}\cdot\boldsymbol{x}_i\right)$. The number of all possible $\boldsymbol{k}$ grows exponentially on dimension d. For illustration, in each examination, we consider a direction of $\boldsymbol{k}$ in the Fourier space, i.e., $\boldsymbol{k} = k\boldsymbol{p}_1$, $\boldsymbol{p}_1$ is a chosen and fixed unit vector, hence $|\boldsymbol{k}| = k$. Then we have $\hat{y}_k = \frac{1}{n}\sum_{i=0}^{n-1} y_i \exp\left(-I2\pi(\boldsymbol{p}_1 \cdot \boldsymbol{x}_j)k\right)$, which is essentially the 1-d Fourier transform of $\{(x_{\boldsymbol{p}_1,i}, y_i)\}_{i=0}^{n-1}$, where $x_{\boldsymbol{p}_1,i} = \boldsymbol{p}_1 \cdot \boldsymbol{x}_i$ is the projection of $\boldsymbol{x}_i$ on the direction $\boldsymbol{p}_1$ [**?**]. For each training dataset, $\boldsymbol{p}_1$ is chosen as the first principle component of the input space. To examine the convergence behavior of different frequency components during the training, we compute the relative difference between the DNN output and the target function for selected important frequencies k's at each recording step, that is, $\Delta_F(k) = |\hat{h}_k - \hat{y}_k|/|\hat{y}_k|$, where $\hat{y}_k$ and $\hat{h}_k$ are 1-d Fourier transforms of $\{y_i\}_{i=0}^{n-1}$ and the corresponding DNN output$\{h_i\}_{i=0}^{n-1}$, respectively, along $\boldsymbol{p}_1$. Note that each response frequency component, $\hat{h}_k$, of DNN output evolves as the training goes.

In the following, we show empirically that the F-Principle is exhibited in the selected direction during the training process of DNNs when applied to MNIST/CIFAR10 with cross-entropy loss. The network for MNIST is a fully-connected tanh DNN (784-400-200-10) and for CIFAR10 is two ReLU convolutional layers followed by a fully-connected DNN (800-400-400-400-10). All experimental details of this paper can be found in Appendix **??**. We consider one of the 10-d outputs in each case using non-uniform Fourier transform. As shown in Fig. 3(a) and 3(c), low frequencies dominate in both real

datasets. During the training, the evolution of relative errors of certain selected frequencies (marked by black squares in Fig. 3(a) and 3(c)) is shown in Fig. 3(b) and 3(d). One can easily observe that DNNs capture low frequencies first and gradually capture higher frequencies. Clearly, this behavior is consistent with the F-Principle. For other components of the output vector and other directions of $\boldsymbol{p}$, similar phenomena are also observed.

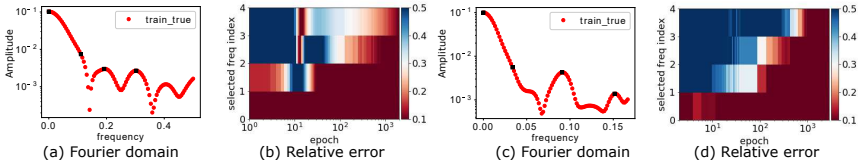


图 3: Projection method. (a, b) are for MNIST, (c, d) for CIFAR10. (a, c) Amplitude $|\hat{y}_k|$ vs. frequency. Selected frequencies are marked by black squares. (b, d) $\Delta_F(k)$ vs. training epochs for the selected frequencies.

3.3.2 Filtering method

The projection method in the previous section enables us to visualize the F-Principle in one direction for each examination at the level of individual frequency components. However, demonstration by this method alone is insufficient because it is impossible to verify the F-Principle at all potentially informative directions for high-dimensional data. To compensate the projection method, in this section, we consider a coarse-grained filtering method which is able to unravel whether, in the radially averaged sense, low frequencies converge faster than high frequencies.

The idea of the filtering method is as follows. We split the frequency domain into two parts, i.e., a low-frequency part with $|\boldsymbol{k}| \leq k_0$ and a high-frequency part with $|\boldsymbol{k}| > k_0$, where $|\cdot|$ is the length of a vector. The DNN is trained as usual by the original dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=0}^{n-1}$, such as MNIST or CIFAR10. The DNN output is denoted as $\boldsymbol{h}$. During the training, we can examine the convergence of relative errors of low- and high- frequency part, using the two measures below

$$e_{\text{low}} = \left(\frac{\sum_{\boldsymbol{k}} \mathbb{1}_{|\boldsymbol{k}|\leq k_0} |\hat{\boldsymbol{y}}(\boldsymbol{k}) - \hat{\boldsymbol{h}}(\boldsymbol{k})|^2}{\sum_{\boldsymbol{k}} \mathbb{1}_{|\boldsymbol{k}|\leq k_0} |\hat{\boldsymbol{y}}(\boldsymbol{k})|^2} \right)^{\frac{1}{2}}, \tag{23}$$

$$e_{\text{high}} = \left(\frac{\sum_{\boldsymbol{k}} (1 - \mathbb{1}_{|\boldsymbol{k}|\leq k_0}) |\hat{\boldsymbol{y}}(\boldsymbol{k}) - \hat{\boldsymbol{h}}(\boldsymbol{k})|^2}{\sum_{\boldsymbol{k}} (1 - \mathbb{1}_{|\boldsymbol{k}|\leq k_0}) |\hat{\boldsymbol{y}}(\boldsymbol{k})|^2} \right)^{\frac{1}{2}}, \tag{24}$$

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

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

If we consistently observe $e_{\text{low}} < e_{\text{high}}$ for different k_0's during the training, then in a mean sense, lower frequencies are first captured by the DNN, i.e., F-Principle.

However, because it is almost impossible to compute above quantities numerically due to high computational cost of high-dimensional Fourier transform, we alternatively use the Fourier transform of a Gaussian function $\hat{G}^\delta(\boldsymbol{k})$, where δ is the variance of the Gaussian function G, to approximate $\mathbb{1}_{|\boldsymbol{k}|>k_0}$. This is reasonable due to the following two reasons. First, the Fourier transform of a Gaussian is still a Gaussian, i.e., $\hat{G}^\delta(\boldsymbol{k})$ decays exponentially as $|\boldsymbol{k}|$ increases, therefore, it can approximate $\mathbb{1}_{|\boldsymbol{k}|\leq k_0}$ by $\hat{G}^\delta(\boldsymbol{k})$ with a proper $\delta(k_0)$ (referred to as δ for simplicity). Second, the computation of e_{low} and e_{high} 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 examination in the spatial domain so as to avoid the almost impossible high-dimensional Fourier transform. The low frequency part can be derived by

$$\boldsymbol{y}_i^{\text{low},\delta} \triangleq (\boldsymbol{y} * G^\delta)_i, \tag{25}$$

where $*$ indicates convolution operator, and the high frequency part can be derived by

$$\boldsymbol{y}_i^{\text{high},\delta} \triangleq \boldsymbol{y}_i - \boldsymbol{y}_i^{\text{low},\delta}. \tag{26}$$

Then, we can examine

$$e_{\text{low}} = \left(\frac{\sum_i |\boldsymbol{y}_i^{\text{low},\delta} - \boldsymbol{h}_i^{\text{low},\delta}|^2}{\sum_i |\boldsymbol{y}_i^{\text{low},\delta}|^2} \right)^{\frac{1}{2}}, \tag{27}$$

$$e_{\text{high}} = \left(\frac{\sum_i |\boldsymbol{y}_i^{\text{high},\delta} - \boldsymbol{h}_i^{\text{high},\delta}|^2}{\sum_i |\boldsymbol{y}_i^{\text{high},\delta}|^2} \right)^{\frac{1}{2}}, \tag{28}$$

where $\boldsymbol{h}^{\text{low},\delta}$ and $\boldsymbol{h}^{\text{high},\delta}$ are obtained from the DNN output $\boldsymbol{h}$, which evolves as a function of training epoch, through the same decomposition. If $e_{\text{low}} < e_{\text{high}}$ for different δ's during the training, F-Principle holds; otherwise, it is falsified. Next, we introduce the experimental procedure.

Step One: **Training**. Train the DNN by the *original dataset* $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=0}^{n-1}$, such as MNIST or CIFAR10. $\boldsymbol{x}_i$ is an image vector, $\boldsymbol{y}_i$ is a one-hot vector.

Step Two: **Filtering**. The low frequency part can be derived by

$$\boldsymbol{y}_i^{\text{low},\delta} = \frac{1}{C_i} \sum_{j=0}^{n-1} \boldsymbol{y}_j G^\delta(\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{29}$$

where $C_i = \sum_{j=0}^{n-1} G^\delta(\boldsymbol{x}_i - \boldsymbol{x}_j)$ is a normalization factor and

$$G^\delta(\boldsymbol{x}_i - \boldsymbol{x}_j) = \exp\left(-|\boldsymbol{x}_i - \boldsymbol{x}_j|^2/(2\delta)\right). \tag{30}$$

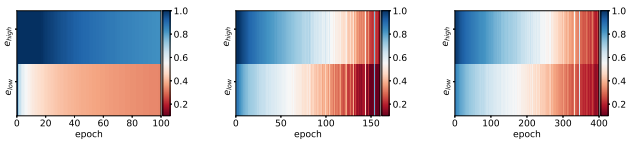
The high frequency part can be derived by $\boldsymbol{y}_i^{\text{high},\delta} \triangleq \boldsymbol{y}_i - \boldsymbol{y}_i^{\text{low},\delta}$. We also compute $\boldsymbol{h}_i^{\text{low},\delta}$ and $\boldsymbol{h}_i^{\text{high},\delta}$ for each DNN output $\boldsymbol{h}_i$.

Step Three: **Examination**. To quantify the convergence of $\boldsymbol{h}^{\text{low},\delta}$ and $\boldsymbol{h}^{\text{high},\delta}$, we compute the relative error e_{low} and e_{high} at each training epoch through Eq. (28).

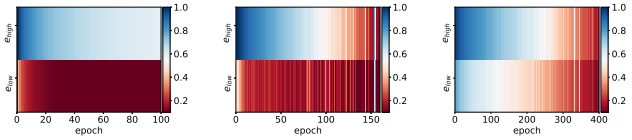
With the filtering method, we show the F-Principle in the DNN training process of real datasets for commonly used large networks. For MNIST, we use a fully-connected tanh-DNN (no softmax) with MSE loss; for CIFAR10, we use cross-entropy loss and two structures, one is small ReLU-CNN network,

i.e., two convolutional layers, followed by a fully-connected multi-layer neural network with a softmax; the other is VGG16 [**?**] equipped with a 1024 fully-connected layer. These three structures are denoted as "DNN", "CNN" and "VGG" in Fig. 4, respectively. All are trained by SGD from *scratch*. More details are in Appendix **??**.

We scan a large range of δ for both datasets. As an example, results of each dataset for several δ's are shown in Fig. 4, respectively. Red color indicates small relative error. In all cases, the relative error of the low-frequency part, i.e., e_{low}, decreases (turns red) much faster than that of the high-frequency part, i.e., e_{high}. Therefore, as analyzed above, the low-frequency part converges faster than the high-frequency part. We also remark that, based on the above results on cross-entropy loss, the F-Principle is not limited to MSE loss, which possesses a natural Fourier domain interpretation by the Parseval's theorem. Note that the above results holds for both SGD and GD.



(a) $\delta = 3$, DNN (b) $\delta = 3$, CNN (c) $\delta = 7$, VGG



(d) $\delta = 7$, DNN (e) $\delta = 7$, CNN (f) $\delta = 10$, VGG

图 4: F-Principle in real datasets. e_{low} and e_{high} indicated by color against training epoch.

4 Theoretical study of F-Principle

A key reason why deep learning is often critized as a black box is the lack of theoretical support. A solid thoery to explain the strength and the weakness of the deep learning is important for better understanding and better usage of the deep learning in practice. The thoery of deep learning resemble other science, such as physics, that is, the theory should be consistent with empirical phenomena and able to predict the behavior of deep learning in real problems. Therefore, it is important to identify universal phenomena which can guide the development of thoeries for deep learning.

The F-Principle, found in both synthetic and real data, qualifies as one phenomenon to induce a underlying theory for deep learning. In this section, we review theories of the F-Principle for various settings. The theories reviewed in this section explores the F-Principle with sufficient large number of training samples. The next section reviews a linear F-Principle with any finite samples, which enables the exploration of the relation between the F-Principle and the generalization.

4.1 Idealized setting

[10, 9] shows an intuitive understanding.

The activation function we consider is $\sigma(x) = \tanh(x)$.

$$\sigma(x) = \tanh(x) = \frac{\mathrm{e}^x - \mathrm{e}^{-x}}{\mathrm{e}^x + \mathrm{e}^{-x}}, \quad x \in \mathbb{R}.$$

For a DNN of one hidden layer with m nodes, 1-d input x and 1-d output:

$$h(x) = \sum_{j=1}^{m} a_j \sigma(w_j x + b_j), \quad a_j, w_j, b_j \in \mathbb{R}, \tag{31}$$

where w_j, a_j, and b_j are called *parameters*, in particular, w_j and a_j are called *weights*, and b_j is also known as a *bias*. In the sequel, we will also use the notation $\theta = \{\theta_{lj}\}$ with $\theta_{1j} = a_j$, $\theta_{2j} = w_j$, and $\theta_{lj} = b_j$, $j = 1, \cdots, m$. Note that $\hat{\sigma}(k) = -\frac{I\pi}{\sinh(\pi k/2)}$ where the Fourier transformation and its inverse transformation are defined as follows:

$$\hat{f}(k) = \int_{-\infty}^{+\infty} f(x)\mathrm{e}^{-Ikx}\,\mathrm{d}x, \quad f(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{f}(k)\mathrm{e}^{Ikx}\,\mathrm{d}k.$$

The Fourier transform of $\sigma(w_j x + b_j)$ with $w_j, b_j \in \mathbb{R}$, $j = 1, \cdots, m$ reads as

$$\widehat{\sigma(w_j \cdot + b_j)}(k) = \frac{2\pi I}{|w_j|}\exp\Big(\frac{Ib_j k}{w_j}\Big)\frac{1}{\exp(-\frac{\pi k}{2w_j}) - \exp(\frac{\pi k}{2w_j})}. \tag{32}$$

Thus

$$\hat{h}(k) = \sum_{j=1}^{m} \frac{2\pi a_j I}{|w_j|}\exp\Big(\frac{Ib_j k}{w_j}\Big)\frac{1}{\exp(-\frac{\pi k}{2w_j}) - \exp(\frac{\pi k}{2w_j})}. \tag{33}$$

We define the amplitude deviation between DNN output and the *target function* $f(x)$ at frequency k as

$$D(k) \triangleq \hat{h}(k) - \hat{f}(k).$$

Write $D(k)$ as $D(k) = A(k)\mathrm{e}^{I\phi(k)}$, where $A(k) \in [0, +\infty)$ and $\phi(k) \in \mathbb{R}$ are the amplitude and phase of $D(k)$, respectively. The loss at frequency k is $L(k) = \frac{1}{2}|D(k)|^2$, where $|\cdot|$ denotes the norm of a complex number. The total loss function is defined as: $L = \int_{-\infty}^{+\infty} L(k)\,\mathrm{d}k$. Note that according to Parseval's theorem, this loss function in the Fourier domain is equal to the commonly used loss of mean squared error, that is, $L = \int_{-\infty}^{+\infty} \frac{1}{2}(h(x) - f(x))^2\,\mathrm{d}x$. For readers' reference, we list the partial derivatives of $L(k)$ with respect to parameters

$$\frac{\partial L(k)}{\partial a_j} = \frac{2\pi}{w_j}\sin\Big(\frac{b_j k}{w_j} - \phi(k)\Big)E_0, \tag{34}$$

$$\frac{\partial L(k)}{\partial w_j} = \left[\sin\Big(\frac{b_j k}{w_j} - \phi(k)\Big)\left(\frac{\pi^2 a_j k}{w_j^3}E_1 - \frac{2\pi a_j}{w_j^2}\right) - \frac{2\pi a_j b_j k}{w_j^3}\cos\Big(\frac{b_j k}{w_j} - \phi(k)\Big)\right]E_0, \tag{35}$$

$$\frac{\partial L(k)}{\partial b_j} = \frac{2\pi a_j b_j k}{w_j^2}\cos\Big(\frac{b_j k}{w_j} - \phi(k)\Big)E_0, \tag{36}$$

where

$$E_0 = \frac{\mathrm{sgn}(w_j)A(k)}{\exp(\frac{\pi k}{2w_j}) - \exp(-\frac{\pi k}{2w_j})},$$
$$E_1 = \frac{\exp(\frac{\pi k}{2w_j}) + \exp(-\frac{\pi k}{2w_j})}{\exp(\frac{\pi k}{2w_j}) - \exp(-\frac{\pi k}{2w_j})}.$$

The descent increment at any direction, say, with respect to parameter θ_{lj}, is

$$\frac{\partial L}{\partial \theta_{lj}} = \int_{-\infty}^{+\infty} \frac{\partial L(k)}{\partial \theta_{lj}} \, \mathrm{d}k. \tag{37}$$

The absolute contribution from frequency k to this total amount at θ_{lj} is

$$\left| \frac{\partial L(k)}{\partial \theta_{lj}} \right| \approx A(k) \exp\left(-|\pi k/2w_j|\right) F_{lj}(\theta_j, k), \tag{38}$$

where $\theta_j \triangleq \{w_j, b_j, a_j\}$, $\theta_{lj} \in \theta_j$, $F_{lj}(\theta_j, k)$ is a function with respect to θ_j and k, which can be found in one of Eqs. (34, 35, 36).

When the component at frequency k where $\hat{h}(k)$ is not close enough to $\hat{f}(k)$, $\exp\left(-|\pi k/2w_j|\right)$ would dominate $G_{lj}(\theta_j, k)$ for a small w_j. Through the above framework of analysis, we have the following theorem. Define

$$W = (w_1, w_2, \cdots, w_m)^T \in \mathbb{R}^m. \tag{39}$$

Theorem 1. *Consider a one hidden layer DNN with activation function $\sigma(x) = \tanh x$. For any frequencies k_1 and k_2 such that $|\hat{f}(k_1)| > 0$, $|\hat{f}(k_2)| > 0$, and $|k_2| > |k_1| > 0$, there exist positive constants c and C such that for sufficiently small δ, we have*

$$\frac{\mu\left(\left\{W : \left|\frac{\partial L(k_1)}{\partial \theta_{lj}}\right| > \left|\frac{\partial L(k_2)}{\partial \theta_{lj}}\right| \quad \text{for all} \quad l, j\right\} \cap B_\delta\right)}{\mu(B_\delta)} \geq 1 - C\exp(-c/\delta), \tag{40}$$

where $B_\delta \subset \mathbb{R}^m$ is a ball with radius δ centered at the origin and $\mu(\cdot)$ is the Lebesgue measure.

We remark that c and C depend on k_1, k_2, $|\hat{f}(k_1)|$, $|\hat{f}(k_2)|$, $\sup|a_i|$, $\sup|b_i|$, and m.

证明. To prove the statement, it is sufficient to show that $\mu(S_{lj,\delta})/\mu(B_\delta) \leq C\exp(-c/\delta)$ for each l, j, where

$$S_{lj,\delta} := \left\{ W \in B_\delta : \left|\frac{\partial L(k_1)}{\partial \theta_{lj}}\right| \leq \left|\frac{\partial L(k_2)}{\partial \theta_{lj}}\right| \right\}. \tag{41}$$

We prove this for $S_{1j,\delta}$, that is, $\theta_{lj} = a_j$. The proofs for $\theta_{lj} = w_j$ and b_j are similar. Without loss of generality, we assume that $k_1, k_2 > 0$, $b_j > 0$, and $w_j \neq 0$, $j = 1, \cdots, m$. According to Eq. (34), the inequality $|\frac{\partial L(k_1)}{\partial a_j}| \leq |\frac{\partial L(k_2)}{\partial a_j}|$ is equivalent to

$$\begin{aligned} &\frac{A(k_2)}{A(k_1)} \left| \frac{\exp(\frac{\pi k_1}{2w_j}) - \exp(-\frac{\pi k_1}{2w_j})}{\exp(\frac{\pi k_2}{2w_j}) - \exp(-\frac{\pi k_2}{2w_j})} \right| \cdot \left| \sin\left(\frac{b_j k_2}{w_j} - \phi(k_2)\right) \right| \\ &\geq \left| \sin\left(\frac{b_j k_1}{w_j} - \phi(k_1)\right) \right| \end{aligned} \tag{42}$$

Note that $|\hat{h}(k)| \le C\sum_{j=1}^{m}\frac{|a_j|}{|w_j|}\exp(-\frac{\pi k}{2|w_j|})$ for $k>0$. Thus

$$\lim_{W\to 0}\hat{h}(k)=0 \quad \text{and} \quad \lim_{W\to 0}D(k)=-\hat{f}(k). \tag{43}$$

Therefore,

$$\lim_{W\to 0}A(k)=|\hat{f}(k)| \quad \text{and} \quad \lim_{W\to 0}\phi(k)=\pi+\arg(\hat{f}(k)). \tag{44}$$

For $W\in B_\delta$ with sufficiently small δ, $A(k_1)>\frac{1}{2}|\hat{f}(k_1)|>0$ and $A(k_2)<2|\hat{f}(k_2)|$. Also note that $|\sin(\frac{b_jk_2}{w_j}-\phi(k_2))|\le 1$ and that for sufficiently small δ,

$$\left|\frac{\exp(\frac{\pi k_1}{2w_j})-\exp(-\frac{\pi k_1}{2w_j})}{\exp(\frac{\pi k_2}{2w_j})-\exp(-\frac{\pi k_2}{2w_j})}\right| \le 2\exp\Big(\frac{-\pi(k_2-k_1)}{2|w_j|}\Big). \tag{45}$$

Thus, inequality (42) implies that

$$\Big|\sin\Big(\frac{b_jk_1}{w_j}-\phi(k_1)\Big)\Big| \le \frac{8|\hat{f}(k_2)|}{|\hat{f}(k_1)|}\exp\Big(-\frac{\pi(k_2-k_1)}{2|w_j|}\Big). \tag{46}$$

Noticing that $\frac{2}{\pi}|x|\le|\sin x|$ ($|x|\le\frac{\pi}{2}$) and Eq. (44), we have for $W\in S_{lj,\delta}$, for some $q\in\mathbb{Z}$,

$$\Big|\frac{b_ik_1}{w_i}-\arg(\hat{f}(k_1))-q\pi\Big| \le \frac{8\pi|\hat{f}(k_2)|}{|\hat{f}(k_1)|}\exp\Big(-\frac{\pi(k_2-k_1)}{2\delta}\Big) \tag{47}$$

that is,

$$\begin{aligned}&-c_1\exp(-c_2/\delta)+q\pi+\arg(\hat{f}(k_1))\\ &\le \frac{b_ik_1}{w_i}\le c_1\exp(-c_2/\delta)+q\pi+\arg(\hat{f}(k_1)),\end{aligned} \tag{48}$$

where $c_1=\frac{8\pi|\hat{f}(k_2)|}{|\hat{f}(k_1)|}$ and $c_2=\pi(k_2-k_1)$. Define $I:=I^+\cup I^-$ where

$$I^+:=\{w_j>0: W\in S_{1j,\delta}\},\quad I^-:=\{w_j<0: W\in S_{1j,\delta}\}. \tag{49}$$

For $w_j>0$, we have for some $q\in\mathbb{Z}$,

$$\begin{aligned}0&<\frac{b_jk_1}{c_1\exp(-c_2/\delta)+q\pi+\arg(\hat{f}(k_1))}\\ &\le w_j\le\frac{b_jk_1}{-c_1\exp(-c_2/\delta)+q\pi+\arg(\hat{f}(k_1))}.\end{aligned} \tag{50}$$

Since $W\in B_\delta$ and $c_1\exp(-c_2/\delta)+\arg(\hat{f}(k_1))\le 2\pi$, we have $\frac{b_jk_1}{2\pi+q\pi}\le w_j\le\delta$. Then Eq. (50) only holds for some large q, more precisely, $q\ge q_0:=\frac{b_jk}{\pi\delta}-2$. Thus we obtain the estimate for the (one-dimensional) Lebesgue measure of I^+

$$\begin{aligned}\mu(I^+)&\le\sum_{q=q_0}^{\infty}\left|\frac{b_jk_1}{-c_1\exp(-c_2/\delta)+q\pi+\arg(\hat{f}(k_1))}\right.\\ &\qquad\left.-\frac{b_jk_1}{c_1\exp(-c_2/\delta)+q\pi+\arg(\hat{f}(k_1))}\right|\\ &\le 2|b_j|k_1c_1\exp(-c_2/\delta)\\ &\quad\cdot\sum_{q=q_0}^{\infty}\frac{1}{(q\pi+\arg(\hat{f}(k_1)))^2-(c_1\exp(-c_2/\delta))^2}\\ &\le C\exp(-c/\delta).\end{aligned} \tag{51}$$

The similar estimate holds for $\mu(I^-)$, and hence $\mu(I) \leq C\exp(-c/\delta)$. For $W \in B_\delta$, the $(m-1)$ dimensional vector $(w_1, \cdots, w_{j-1}, w_{j+1}, \cdots, w_m)^T$ is in a ball with radius δ in $\mathbb{R}^{m-1}$. Therefore, we final arrive at the desired estimate

$$\frac{\mu(S_{1j,\delta})}{\mu(B_\delta)} \leq \frac{\mu(I)\omega_{m-1}\delta^{m-1}}{\omega_m\delta^m} \leq C\exp(-c/\delta), \tag{52}$$

where ω_m is the volume of a unit ball in $\mathbb{R}^m$. □

Theorem 2. *Considering a DNN of one hidden layer with activation function $\sigma(x) = \tanh(x)$. Suppose the target function has only two non-zero frequencies k_1 and k_2, that is, $|\hat{f}(k_1)| > 0$, $|\hat{f}(k_2)| > 0$, and $|k_2| > |k_1| > 0$, and $|\hat{f}(k)| = 0$ for $k \neq k_1, k_2$. Consider the loss function of $L = L(k_1) + L(k_2)$ with gradient descent training. Denote*

$$\mathcal{S} = \left\{ \frac{\partial L(k_1)}{\partial t} \leq 0, \frac{\partial L(k_1)}{\partial t} \leq \frac{\partial L(k_2)}{\partial t} \right\},$$

that is, $L(k_1)$ decreases faster than $L(k_2)$. There exist positive constants c and C such that for sufficiently small δ, we have

$$\frac{\mu\left(\{W : \mathcal{S} \quad \text{holds}\} \cap B_\delta\right)}{\mu(B_\delta)} \geq 1 - C\exp(-c/\delta),$$

where $B_\delta \subset \mathbb{R}^m$ is a ball with radius δ centered at the origin and $\mu(\cdot)$ is the Lebesgue measure.

证明. By gradient descent algorithm, we obtain

$$\begin{aligned}
\frac{\partial L(k_1)}{\partial t} &= \sum_{l,j} \frac{\partial L(k_1)}{\partial \theta_{lj}} \frac{\partial \theta_{lj}}{\partial t} \\
&= -\sum_{l,j} \frac{\partial L(k_1)}{\partial \theta_{lj}} \frac{\partial (L(k_1) + L(k_2))}{\partial \theta_{lj}} \\
&= -\sum_{l,j} \left(\frac{\partial L(k_1)}{\partial \theta_{lj}} \right)^2 - \sum_{l,j} \frac{\partial L(k_1)}{\partial \theta_{lj}} \frac{\partial L(k_2)}{\partial \theta_{lj}}, \\
\frac{\partial L(k_2)}{\partial t} &= -\sum_{l,j} \left(\frac{\partial L(k_2)}{\partial \theta_{lj}} \right)^2 - \sum_{l,j} \frac{\partial L(k_1)}{\partial \theta_{lj}} \frac{\partial L(k_2)}{\partial \theta_{lj}},
\end{aligned}$$

and

$$\frac{\partial L}{\partial t} = \frac{\partial \left(L(k_1) + L(k_2)\right)}{\partial t} = -\sum_{l,j} \left(\frac{\partial L(k_1)}{\partial \theta_{lj}} + \frac{\partial L(k_2)}{\partial \theta_{lj}} \right)^2 \leq 0. \tag{53}$$

To obtain

$$0 > \frac{\partial L(k_1)}{\partial t} - \frac{\partial L(k_2)}{\partial t} = -\sum_{l,j} \left[\left(\frac{\partial L(k_1)}{\partial \theta_{lj}} \right)^2 - \left(\frac{\partial L(k_2)}{\partial \theta_{lj}} \right)^2 \right], \tag{54}$$

it is sufficient to have

$$\left| \frac{\partial L(k_1)}{\partial \theta_{lj}} \right| > \left| \frac{\partial L(k_2)}{\partial \theta_{lj}} \right|. \tag{55}$$

Eqs. (53, 54) also yield to

$$\frac{\partial L(k_1)}{\partial t} < 0.$$

Therefore, Eq. (55) is a sufficient condition for $\mathcal{S}$. Based on the theorem 1, we have proved the theorem 2. □

5 Reading

Chaper 5 of Deep Learning [2], http://www.deeplearningbook.org/
Suggested Notation for Machine Learning [8].
Prof. Yanyang Xiao's fft note.

参考文献

[1] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[3] 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. *arXiv preprint arXiv:1806.08734*, 2018.

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

[5] Lei Wu, Zhanxing Zhu, and Weinan E. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

[6] Zhi-Qin J Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. *arXiv preprint arXiv:1807.01251*, 2018.

[7] Zhi-Qin John Xu. Frequency principle in deep learning with general loss functions and its potential application. *arXiv preprint arXiv:1811.10146*, 2018.

[8] Zhi-Qin John Xu, Tao Luo, Zheng Ma, and Yaoyu Zhang. Suggested notation for machine learning. 2020. `https://github.com/Mayuyu/suggested-notation-for-machine-learning`.

[9] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

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

[11] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.