Frequency Principle

Yaoyu Zhang, Tao Luo
• Background

• Frequency Principle

• Implication of F-Principle

• Quantitative theory for F-Principle
Background—Why DNN remains a mystery?
Supervised Learning Problem

Given $\mathcal{D}: \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}: \{f(\cdot; \Theta) | \Theta \in \mathbb{R}^m\}$, find $f \in \mathcal{H}$ such that $f(x_i) = y_i$ for $i = 1, \ldots, n$.

Example 1 (mystery)—Deep Learning

\[
f_\theta(x) = W^{[L]}\sigma \circ (\cdots W^{[2]}\sigma \circ (W^{[1]}x + b^{[1]}) + \cdots) + b^{[L]}
\]

\[
\dot{\Theta} = -\nabla_{\Theta} L(\Theta)
\]

Initialized by special $\Theta_0$

\[
L(\Theta) = \Sigma_{i=1}^n(h(x_i; \Theta) - y_i)^2 / 2n
\]
Supervised Learning Problem

Given $\mathcal{D}: \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{H}: \{f(\cdot; \Theta) | \Theta \in \mathbb{R}^m\}$, find $f \in \mathcal{H}$ such that $f(x_i) = y_i$ for $i = 1, \ldots, n$.

Example 2 (well understood)—polynomial interpolation

$\mathcal{D}$

$\{(x_i \in \mathbb{R}, y_i \in \mathbb{R})\}_{i=1}^n$

$\mathcal{H}$

$h(x; \Theta) = \theta_1 + \cdots + \theta_M x^{m-1}$

with $m = n$

find

Newton's interpolation formula

Q: Why we think polynomial interpolation is well understood?
Supervised Learning Problem

Given $\mathcal{D}: \{(x_i, y_i)\}_{i=1}^{n}$ and $\mathcal{H}: \{f(\cdot; \Theta) | \Theta \in \mathbb{R}^m\}$, find $f \in \mathcal{H}$ such that $f(x_i) = y_i$ for $i = 1, \ldots, n$.

Example 3 (well understood)—linear spline

$\mathcal{D}$

$\{(x_i \in \mathbb{R}, y_i \in \mathbb{R})\}_{i=1}^{n}$

$\mathcal{H}$

piecewise linear functions

find

explicit solution
Why deep learning is a mystery?

Given $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^{n}$ and $\mathcal{H} := \{f(\cdot; \Theta) | \Theta \in \mathbb{R}^m\}$, find $f \in \mathcal{H}$ such that $f(x_i) = y_i$ for $i = 1, \ldots, n$.

### Deep learning (black box!)

- **$\mathcal{D}$**
  - High dimensional real data (e.g., $d=32*32*3$)

- **$\mathcal{H}$**
  - Deep neural network (#para $\gg$ #data)

- **find**
  - Gradient-based method with proper initialization

### Conventional methods

- **Low dimensional data ($d \leq 3$)**

- **Spanned by simple basis functions ($#_{\text{para}} \leq #_{\text{data}}$)**

- **explicit formula**
Is deep learning alchemy?
Golden ages of neural network

1960-1969
- Simple (#data small)
- Single-layer NN (cannot solve XOR)
- Non-Gradient based (nondiff activation)

1984-1996
- Moderate (e.g., MNIST)
- Multi-layer NN (universal approx)
- Gradient based (BP)

2010-now
- Complex real data (e.g., ImageNet)
- Deep NN
- Gradient based (BP) with good initialization

NN is still a black box!
1. Why don’t heavily parameterized neural networks overfit the data?

2. What is the effective number of parameters?

3. Why doesn’t backpropagation head for a poor local minima?

4. When should one stop the backpropagation and use the current parameters?
Frequency Principle
Conventional view of generalization
Conventional view of generalization

"With four parameters you can fit an elephant to a curve; with five you can make him wiggle his trunk.”

-- John von Neumann

A model that can fit anything likely overfits the data.

Mayer et al., 2010
Overparameterized DNNs often generalize well.
Problem simplification

\[ \mathcal{D} \]

\[ \mathcal{H} \]

\[ f_\theta(x) = W_1 \sigma \circ (\cdots W_2 \sigma \circ (W_1 x + b_1) + \cdots) + b \]

\[ \text{find } \dot{\Theta} = -\nabla_\Theta L(\Theta) \]

 Initialized by special \( \Theta_0 \)

only observe \( f(x, t) = f(x; \Theta(t)) \)
Overparameterized DNNs still generalize well

Lei Wu, Zhanxing Zhu, Weinan E, 2017

#para(~1000)>>#data: 5
evolution of $f(x, t)$

tanh-DNN, 200-100-100-50
Through the lens of Fourier transform $\hat{h}(\xi, t)$

**Frequency Principle (F-Principle):**

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

Xu, Zhang, Xiao, *Training behavior of deep neural network in frequency domain*, 2018
Synthetic curve with equal amplitude
How DNN fits a 2-d image?

Target: image $I(x): \mathbb{R}^2 \rightarrow \mathbb{R}$

$x$: location of a pixel

$I(x)$: grayscale pixel value

(a) True image

(b) DNN output
High-dimensional real data?

Xu, Zhang, Luo, Xiao, Ma, Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks, 2019
Frequency

Image frequency (NOT USED)

- This frequency corresponds to the rate of change of intensity across neighboring pixels.

Response frequency

- Frequency of a general Input-Output mapping $f$.

$$\hat{f}(k) = \int f(x)e^{-i2\pi k \cdot x} \, dx$$

**MNIST:** $\mathbb{R}^{784} \rightarrow \mathbb{R}^{10}, k \in \mathbb{R}^{784}$

Zero freq

Same color

high freq

Sharp edge

high freq

Adversarial example

Goodfellow et al.
Examining F-Principle for high dimensional real problems

Nonuniform Discrete Fourier transform (NUDFT) for training dataset \{((x_i, y_i))\}_{i=1}^n:

\[ \hat{y}_k = \frac{1}{n} \sum_{i=1}^n y_i e^{-i2\pi k \cdot x_i}, \quad \hat{h}_k(t) = \frac{1}{n} \sum_{i=1}^n h(x_i, t) e^{-i2\pi k \cdot x_i} \]

Difficulty:

- Curse of dimensionality, i.e., \#k grows exponentially with dimension of problem \(d\).

Our approaches:

- **Projection**, i.e., choose \(k = kp_1\)
- **Filtering**
Projection approach

Relative error: \( \Delta_F(k) = \left| \hat{h}_k - \hat{y}_k \right| / |\hat{y}_k| \)
Decompose frequency domain by filtering

\[ y_{i}^{\text{low},\delta} = (y * G^\delta)_i \]

\[ y_{i}^{\text{high},\delta} \triangleq y_i - y_{i}^{\text{low},\delta} \]

\[ e_{\text{low}} = \left( \frac{\sum_i |y_{i}^{\text{low},\delta} - h_{i}^{\text{low},\delta}|^2}{\sum_i |y_{i}^{\text{low},\delta}|^2} \right)^{\frac{1}{2}} \]

\[ e_{\text{high}} = \left( \frac{\sum_i |y_{i}^{\text{high},\delta} - h_{i}^{\text{high},\delta}|^2}{\sum_i |y_{i}^{\text{high},\delta}|^2} \right)^{\frac{1}{2}} \]
F-Principle in high-dim space

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

(d) $\delta = 7$, DNN
(e) $\delta = 7$, CNN
(f) $\delta = 10$, VGG
Implication of F-Principle
Why don’t heavily parameterized neural networks overfit the data?

F-Principe: DNN prefers low frequencies

For \( \bar{x} \in \{-1,1\}^n \)

\[
f(\bar{x}) = \prod_{j=1}^{n} x_j,
\]

Even \#‘-1’ \(\rightarrow\) 1;
Odd \#‘-1’ \(\rightarrow\) -1.

Test accuracy: 72% \(\gg\) 10%
Test accuracy: \(~50\%,\) random guess
When should one stop the backpropagation and use the current parameters?
Studies elicited by F-Principle

• **Theoretical study**

• **Empirical study**
  - Rabinowitz, N. C. (2019), ‘Meta-learners’ learning dynamics are unlike learners”,

• **Application**
Quantitative theory for F-Principle

Zhang, Xu, Luo, Ma, Explicitizing an Implicit Bias of the Frequency Principle in Two-layer Neural Networks, 2019
The NTK regime

\[ L(\Theta) = \sum_{i=1}^{n} (h(x_i; \Theta) - y_i)^2 \]
\[ \dot{\Theta} = -\nabla_{\Theta} L(\Theta) \]

- \( \partial_t h(x; \Theta) = -\sum_{i=1}^{n} K_{\Theta}(x, x_i)(h(x_i; \Theta) - y_i) \)

Where \( K_{\Theta}(x, x') = \nabla_{\Theta} h(x; \Theta) \cdot \nabla_{\Theta} h(x'; \Theta) \)

- Neural Tangent Kernel (NTK) regime:
  \( K_{\Theta(t)}(x, x') \approx K_{\Theta(0)}(x, x') \) for any \( t \).

**Theorem 1.** For a network of depth \( L \) at initialization, with a Lipschitz nonlinearity \( \sigma \), and in the limit as the layers width \( n_1, \ldots, n_{L-1} \to \infty \) sequentially, the NTK \( \Theta^{(L)} \) converges in probability to a deterministic limiting kernel:

\[ \Theta^{(L)} \to \Theta_{\infty}^{(L)} \otimes Id_{n_L}. \]

Jacot et al., 2018
Problem simplification

\( D \)

\( \mathcal{H} \)

\[ f_\theta(x) = W^{[L]} \sigma \circ (\cdots W^{[2]} \sigma \circ (W^{[1]} x + b^{[1]} + \cdots) + b^{[L]}) \]

\textbf{find} \quad \dot{\Theta} = -\nabla_\Theta L(\Theta) \quad \text{Initialized by special } \Theta_0

Two-layer ReLU NN

\[ h(x; \Theta) = \sum_{i=1}^{n} w_i \sigma (r_i (x + l_i)) \]

Kernel gradient flow

\[ \partial_t f(x, t) = -\sum_{i=1}^{n} K_{\Theta_0}(x, x_i)(f(x_i, t) - y_i) \]
Linear F-Principle (LFP) dynamics

2-layer NN: $h(x; \Theta) = \sum_{i=1}^{n} w_i \text{ReLU}(r_i(x + l_i))$

Assumptions:
(i) NTK regime, (ii) sufficiently wide distribution of $l_i$.

$$\partial_t \hat{h}(\xi, t) = - \left[ \frac{4\pi^2 \langle r^2 w^2 \rangle}{\xi^2} + \frac{\langle r^2 \rangle + \langle w^2 \rangle}{\xi^4} \right] \left( \hat{h}_p(\xi, t) - \hat{f}_p(\xi, t) \right)$$

$\langle \cdot \rangle$: mean over all neurons at initialization
$f$: target function; $\langle \cdot \rangle_p = \langle \cdot \rangle_p$, where $p(x) = \frac{1}{n} \sum_{i=1}^{n} \delta(x - x_i)$;
$\hat{\cdot}$: Fourier transform; $\xi$: frequency

aliasing
Preference induced by LFP dynamics

\[ \partial_t \hat{h}(\xi, t) = -\left[ \frac{4\pi^2 \langle r^2 w^2 \rangle}{\xi^2} + \frac{\langle r^2 \rangle + \langle w^2 \rangle}{\xi^4} \right] \left( \hat{h}_p(\xi, t) - \hat{f}_p(\xi, t) \right) \]

\[ \min_{\xi \in F_{\gamma}} \int \left[ \frac{4\pi^2 \langle r^2 w^2 \rangle}{\xi^2} + \frac{\langle r^2 \rangle + \langle w^2 \rangle}{\xi^4} \right]^{-1} |\hat{h}(\xi)|^2 \, d\xi \]

s.t. \( h(x_i) = y_i \) for \( i = 1, \ldots, n \)

Case 1: \( \xi^{-2} \) dominant
- \( \min \int \xi^2 |\hat{h}(\xi)|^2 \, d\xi \sim \min \int |h'(x)|^2 \, d\xi \rightarrow \text{linear spline} \)

Case 2: \( \xi^{-4} \) dominant
- \( \min \int \xi^4 |\hat{h}(\xi)|^2 \, d\xi \sim \min \int |h''(x)|^2 \, d\xi \rightarrow \text{cubic spline} \)
Regularity can be changed through initialization

**Case 1**

\[
\langle r^2 \rangle + \langle w^2 \rangle \gg 4\pi^2 \langle r^2 w^2 \rangle
\]

\[
\min \int \xi^2 |\hat{h}(\xi)|^2 \, d\xi
\]

**Case 2**

\[
4\pi^2 \langle r^2 w^2 \rangle \gg \langle r^2 \rangle + \langle w^2 \rangle
\]

\[
\min \int \xi^4 |\hat{h}(\xi)|^2 \, d\xi
\]
High-dimensional Case

\[ \partial_t \hat{h}(\xi, t) = - \left[ \frac{\langle |r|^2 \rangle + \langle w^2 \rangle}{|\xi|^{d+3}} + \frac{4\pi^2 \langle |r|^2 w^2 \rangle}{|\xi|^{d+1}} \right] (\hat{h}_p(\xi, t) - \hat{f}_p(\xi, t)) \]

where \( f \): target function; \( (\cdot)_p = (\cdot)p \), where \( p(x) = \frac{1}{n} \sum_{i=1}^{n} \delta(x - x_i) \); \( \cdot \): Fourier transform; \( \xi \): frequency.

**Theorem (informal).** Solution of LFP dynamics at \( t \to \infty \) with initial value \( h_{ini} \) is the same as solution of the following optimization problem

\[
\min_{h - h_{ini} \in F_Y} \int \left[ \frac{\langle |r|^2 \rangle + \langle w^2 \rangle}{|\xi|^{d+3}} + \frac{4\pi^2 \langle |r|^2 w^2 \rangle}{|\xi|^{d+1}} \right]^{-1} |\hat{h}(\xi) - \hat{h}_{ini}(\xi)|^2 \, d\xi
\]

s.t. \( h(X) = Y \).
**FP-norm and FP-space**

We define the FP-norm for all function $h \in L^2(\Omega)$:

$$
\|h\|_\gamma = \|\hat{h}\|_{H^\Gamma} = \left(\sum_{k \in \mathbb{Z}^{d*}} \gamma^{-2}(k) |\hat{h}(k)|^2\right)^{1/2}
$$

Next, we define the FP-space:

$$
F_\gamma(\Omega) = \{h \in L^2(\Omega): \|h\|_\gamma < \infty\}
$$

**A priori generalization error bound**

**Theorem (informal).** Suppose that the real-valued target function $f \in F_\gamma(\Omega)$, $h_n$ is the solution of the regularized model

$$
\min_{h \in F_\gamma} \|h\|_\gamma \text{ s.t. } h(X) = Y
$$

Then for any $\delta \in (0,1)$ with probability at least $1 - \delta$ over the random training samples, the population risk has the bound

$$
L(h_n) \leq (\|f\|_\infty + 2\|f\|_\gamma \|\gamma\|_{l^2}) \left(\frac{2}{\sqrt{n}} + 4 \sqrt{\frac{2\log(4/\delta)}{n}}\right)
$$
1. Why don’t heavily parameterized neural networks overfit the data?

2. What is the effective number of parameters?

3. Why doesn’t backpropagation head for a poor local minima?

4. When should one stop the backpropagation and use the current parameters?
A picture for the generalization mystery of DNN

- Unbiased initialization
- + F-Principle
- + \( \cdots \)
- (to be discovered)

Global Minima
Conclusion

DNNs prefer low frequencies!

References:

• Xu, Zhang, Xiao, *Training behavior of deep neural network in frequency domain*, 2018

• Xu, Zhang, Luo, Xiao, Ma, *Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks*, 2019

• Zhang, Xu, Luo, Ma, *Explicitizing an Implicit Bias of the Frequency Principle in Two-layer Neural Networks*, 2019

• Zhang, Xu, Luo, Ma, *A type of generalization error induced by initialization in deep neural networks*, 2019