FLT Seminar Series[1], Session 4

Feature Learning Theory for Image Data

Chen Yanbo[2]

Ph.D. Candidate
School of Computer Science, Wuhan University
430072, Wuhan, Hubei, China

Jun. 15st, 2025

[1] This project is open for collaboration. For details, see our project page at https://github.com/yanboc/feature-learning-theory.

[2] Contact: yanboch@126.com.

Outline

How FLT Characterizes Vision Tasks?

- Highlights from our last sessions
  - FLT for sequential data (Session 3)
  - The *methodology* of FLT (Sessions 2 & 3)
- Yet another simplified example: **how** FLT characterize image data?
  - How to characterize image classification tasks.
  - A simplified FLT analysis setting for sequential tasks.

Table of Contents

How FLT Characterizes sequential tasks?

The 4 core elements for sequential tasks

- *Data.* Token sequences $(v_1, v_2, \cdots, v_t)$, where the tokens $\boldsymbol{v}_i$ $(i \in [t])$ are chosen from an (abstract) vocabulary set $\mathcal{V}$.
- *Hypothesis classes.* ***Transformers (TFs)*** and others (e.g., mamba)
  - Transformer Explainer (which help me better understand TFs)
  - **Major simplifications**: $x \in \mathbb{R}^d$ as raw input, only one attention head (often integrated with MLP and LM heads)
- *Algorithm.* Similar to other topics. **FLT mostly focuses on GD.**
- *Evaluation*[a]. It is task-specific, e.g.,
  - Sentiment Analysis: token sequence $\longrightarrow$ class label
  - Machine Translation: token sequence $\longrightarrow$ token sequence
  - Generation: token sequence $\longrightarrow$ next token (collapse to classification)

(※) most tasks can be characterized as *classification or regression.*

[a] Including generalization ability, robustness, etc. For simplicity, we only discuss accuracy (w.r.t. specific loss function) on *training data.*

The methodology of FLT (Sessions 2 & 3)

- **Specify everything** - *what is the problem to solve?*
  - Create a "virtual environment" (theoretical framework, assumptions, parameters, ...) to perform further analysis.
  - There is a trade-off between triviality & tractability.
  - The no-free-lunch theorem in FLT (Session 3)
  - specified setting + existing proof techniques = determinate results.
- **Seek symmetry** - *how to solve the problem?*
  - Existing proof techniques (*Delayed to Session 5!*)
  - The symmetric structures (e.g., self-similarities of GD steps and the orthonormal assumption of $\boldsymbol{M}$) in FLT frameworks make it sufficient to analyze a single part of a symmetric system rather than all the parts.
- **Programmatic thinking** -*what defines a "good" problem?*
  - Find definitions and simplify them according to real-world practice.
  - There are many parameters in the analysis that require careful tuning.

Table of Contents

A brief introduction to vision tasks

The 4 core elements for vision tasks

- *Data.* **Structured pixel matrices $z \in \mathbb{R}^{C \times d^2}$** (channels $\in \{1, 3\}$)
- *Hypothesis class.* CNNs, ViTs[a], and others.
- *Algorithm.* Similar to other topics. **FLT mostly focuses on GD.**
- *Evaluation.* Variants of CE and MSE losses, or essentially,
  - compare the generated matrix with ground-truth matrix *(Regression)*
  - calculated weighted possibility score for different classes *(Classification)*

[a] Vision Transformers provably learn spatial structure, Jelassi et al., NeurIPS 2022 (https://arxiv.org/abs/2210.09221)

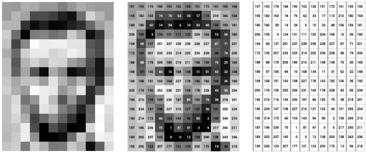

Figure: Image as structured pixel matrix, one channel.

Feature learning intuition

Recall the **no-free-lunch theorem** in FLT:

If a type of network structure performs well on a certain class of data, then it necessarily pays for that with degraded performance on other classes of data.

Why convolutional networks perform well on image data?

- The “tokens” in the images are the **patches** (pixel $\rightarrow$ characters)
- The conv. ops. ($*$) with *stride* $= 1^2$, *kernel size* $= 3^2$, *and no paddings*

$$\boldsymbol{z} * \boldsymbol{w} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 0 & 3 \end{bmatrix}, \tag{1}$$

  which assigns higher scores to *specific patterns* (e.g, edges and shapes).
- The features are stored in the convolution kernels.

How FLT characterizes image data and convolution?

P-patch data model [Allen-Zhu & Li, ICLR 2023]

- The *input image* is characterized as $\boldsymbol{z} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_P) \in \mathbb{R}^{d \times P}$, in which $\boldsymbol{x}_i \in \mathbb{R}^d$ called the ***patches*** of $\boldsymbol{z}$.
- The *convolution* is characterized as inner product $\boldsymbol{x} * \boldsymbol{w} = \langle \boldsymbol{x}, \boldsymbol{w} \rangle$, where the convolution kernels $\boldsymbol{w} \in \mathbb{R}^d$. (stride = kernel size, no paddings)
- (※) The internal structures of $\boldsymbol{x}$ and $\boldsymbol{w}$ are neglected.

$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$
$\boldsymbol{x}_5$	$\boldsymbol{x}_6$	$\boldsymbol{x}_7$	$\boldsymbol{x}_8$
$\boldsymbol{x}_9$	$\boldsymbol{x}_{10}$	$\boldsymbol{x}_{11}$	$\boldsymbol{x}_{12}$
$\boldsymbol{x}_{13}$	$\boldsymbol{x}_{14}$	$\boldsymbol{x}_{15}$	$\boldsymbol{x}_{16}$

Figure: An example of P-patch data ($P = 16$). The 7-th patch is highlighted.

The multi-view data assumption

Question: What is the distribution of the patches $\boldsymbol{x}_{[P]}$ in $\boldsymbol{z}$?

Multi-view data assumption

- Features $\boldsymbol{v}_{[d]}$ are characterized as orthonormal basis in $\mathbb{R}^d$.
- Intuition: in each class (e.g., in cat vs. vehicle classification), some features are essential for classification, and others are auxiliary.
- A given patch $\boldsymbol{x}_i$ ($i \in [P]$) can be characterized as the combination of a feature $\boldsymbol{v}_j$ ($j \in [d]$) and a noise vector ξ. ($\boldsymbol{x}_i$ *contains* $\boldsymbol{v}_j$)
- The distribution of $\boldsymbol{z} \longrightarrow$ random sampling from $\boldsymbol{v}_{[d]}$.

Figure: Illustration of images with multiple views [Allen-Zhu & Li, ICLR 2023].

The Philosophy of FLT

Philosophy No.5., Controlling the randomness of the system

FLT introduces randomness to *enrich the expressiveness* of the theoretical framework at initialization, while the rest of the analysis is *determinate*.

- Data assumption: $\boldsymbol{z} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_P)$ with $\boldsymbol{x}_i = \boldsymbol{v}_j + \xi$.
  - The probability of "$\boldsymbol{z}$ contains $\boldsymbol{v}_j$" for any $j \in [d]$ is fixed. **Essential features is assigned with high probability**.
  - The order of $\boldsymbol{v}_j$ in $\boldsymbol{z}$ is not essential.
  - The size of ξ is relatively small compared to $\boldsymbol{v}_i$.
- Network initialization, mostly from Gaussian distribution. (Making use of the concentration inequalities, delayed to Session 5).

For simplicity, let $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ be the essential vectors for a binary classification.

Hypothesis Class (i.e., Network Structure)

learning goal: predict the label of x

$\Downarrow$

Extract the signal from $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$

A common setting in most analysis

Consider the CNN as follows

$$f_t(x; \boldsymbol{w}^{(t)}) = \sum_{k=1}^{m} \sum_{i=1}^{P} \mathrm{ReLU}\left(\langle \boldsymbol{w}_i^{(t)}, \boldsymbol{x}_i \rangle - b_k^{(t)}\right) \tag{2}$$

The convolution kernels would assign higher score to $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ in the data.

Thanks for your participation!

Welcome to join our WeChat group!
If this expires, please don't hesitate to contact me at yanboch@126.com.