FLT Seminar Series, Session 1

An Introduction to *Feature Learning Theory*

Chen Yanbo [1]

Wuhan University

Summer 2025

[1] Contact: yanboch@126.com.

Outline

An Introduction to Feature Learning Theory (FLT)

- Overview: **what** is *feature*, *learning*, and *theory*, respectively?
  - ***Features*** in various fields (e.g., DS, ML, CS, SE, DL, LLM)
  - How machine ***learns***? The *paradigm* of modern ML&DL
  - Statistical learning ***theory*** v.s. FLT
- A simplified example: **how** FLT works? *(Delayed to Session 2!)*
  - Theoretical framework: creating the “virtual environment”
  - Typical techniques and proof sketches.
- Future plan (Jun. 2025 - Apr. 2026)
  - Seminars (topics and techniques)
  - Projects (GitHub repo + the survey paper)

Table of Contents

Features in various fields

In general, a ***feature*** refers to a *property* or *characteristic* of the object being observed. For example,

Example (Feature in various fields)

- In DS and (classic) ML, features mainly refer to the ***characteristics of the data*** (as the input of the model). ➔ "*Feature Engineering*"
- In CS and SE, the meaning of features depends on specific context. ➔ "*XXX is not a bug, it is a feature.*"
- (⋇) In DL and LLM, roughly speaking, higher-level feature ≈ *data* ***representation***, lower-level feature ≈ *data* ***pattern***.
  - after one convolutional layer ➔ pattern (e.g., edges and shapes)
  - after many layers ➔ representation (used for classification)

Features in various fields

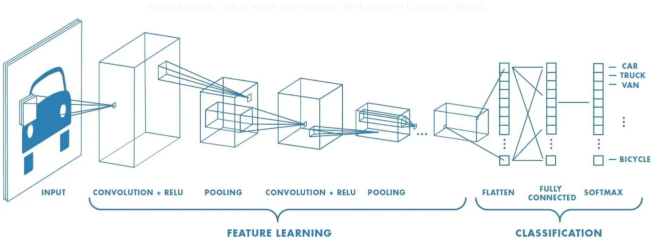

Figure: Higher- and lower-level features in CNN-based classification

Remark

1. We focus on the features in DL (on *data, NNs, and specific tasks*).
2. Feature can be **easily misused** due to its inherent *ambiguity*.
3. The main goal of ML&DL is to ***learn*** **useful features** from data.

How machine learns?

The main goal of ML&DL is to ***learn*** **useful features** from data

$\Uparrow$

Looks good? But **not rigorous (specific) enough**!

The **paradigm** of machine learning

- **Data**. The ground-truth; ➔ $\mathbb{R}^d$ or other structural data
- **Hypothesis class**. A finite set of candidate models ➔ parameterized NNs, e.g., $f(\boldsymbol{x}; \boldsymbol{w})$ with $\boldsymbol{w} \in \mathbb{R}^{depth \times width}$
- **Algorithm**. How to find a good model from the hypothesis class?
- **Evaluation**. What defines a *task-specific* **good** model? ***Loss***!

cf. **Understand Machine Learning**, *S. Shalev-Shwartz and S. Ben-David*

How machine learns? ***Theory v.s. Practice***

Table: The paradigm of modern machine learning

	Theoretical	**In Practice**
Data	vectors and matrices	tensor
Hypothesis Class	functions and mappings	multi-layer NN
Algorithm	optimization	optimizer, LR, ...
Evaluation	loss function, regularization	CE, MSE, ...

```python
batch_size = 64
train_dataloader = DataLoader(training_data, batch_size=batch_size)
class NeuralNetwork(nn.Module):
    def __init__(self):
        ...

    def forward(self, x):
        ...

loss_fn = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=1e-3)

def train(dataloader, model, loss_fn, optimizer):
    ...
    for batch, (X, y) in enumerate(dataloader):
        # Compute prediction error
        pred = model(X)
        loss = loss_fn(pred, y)
        # Backpropagation
        loss.backward()
        optimizer.step()
```

Figure: Typical training code (from PyTorch documentation)

In summary, **machine learning** uses a specified ***algorithm*** to find the best model in the ***hypothesis class*** according to the performance of the model on the **data**, concerning the **evaluation** standard.

Remark

1. Currently, ML theory falls far behind practice.
2. Learning theory topics focus on *separated* stages in the paradigm.
3. **Paradigm shifting**: RL, in-context learning, and inference of LLM.

Statistical learning theory v.s. FLT?

Table: Some learning theory topics

Topic	Idea
Probably Approximately Correct (PAC)	sample complexity
VC Dimension	capacity measure
Expressiveness	universal approximation
Online-Learning	regret minimization
Robust Optimization	minimax, DRO
Diffusion Models	SDE
Feature Learning Theory	training dynamics
...	...

(Wait so... What to do with the features?)

Statistical learning theory v.s. FLT

In summary, **machine learning** uses a specified ***algorithm*** to find the best model in the ***hypothesis class*** according to the performance of the model on the **data**, concerning the **evaluation** standard.

$\Downarrow$

FLT *specifies* the learning task (**network structure**, **data assumption**, **loss**, and **algorithm**) and explore the ***dynamics*** of training.

$\Downarrow$

Dynamics: how the parameters of the NN iterate from random initialization (noise) to useful features used for classification/regression?

Table of Contents

How FLT works?

FLT *specifies* the learning task (**network structure**, **data assumption**, **loss**, and **algorithm**)

$\Downarrow$

Theoretical framework

- **Network Structure**: 2-layer (symmetric)-ReLU $f(x; w)$
- **Data**: orthogonal feature + sparse coding model

$$x = \boldsymbol{M}z + \xi, \; y = sign(\langle w^*, z \rangle)$$

- **Algorithm**: GD with random initialization

$$w^{(t+1)} = w - \eta \nabla Loss(f(x, w), y)$$

- **Loss**: mean square loss + logistic loss

For any specific task, create a **"virtual environment"** to play around with.

Techniques

- Define "**good neuron sets**" (lottery ticket theorem)
- Mathematical induction

(We will elaborate on these in FLT-2)

Table of Contents

Future Plan

GitHub Repo

- Tutorials (in Jupyter notebook) 👍
- Python module (FLTK)

The survey paper

- Literature review and seminar speaker 👍
- Open to cooperation!

Visit our project page at
https://github.com/yanboc/feature-learning-theory

Thanks for your participation!

Welcome to join our WeChat group!

If this expires, please don't hesitate to contact me at `yanboch@126.com`.