Feature Learning by Least Generalization

Hien D. Nguyen [1,2] hiennd@uit.edu.vn & **Chiaki Sakama** [3] sakama@wakayama-u.ac.jp

University of Information Technology, Ho Chi Minh city, Vietnam[1]; Vietnam National University, Ho Chi Minh city, Vietnam[2]; Wakayama University, Wakayama, Japan[3]

Abstract: This paper provides an empirical study for feature learning based on induction. We encode image data into first-order expressions and compute their *least generalization*. An interesting question is whether the least generalization can extract a common pattern of input data. We also introduce two different methods for feature extraction based on symbolic manipulation. We perform experiments using the MNIST datasets and show that the proposed methods successfully capture features from training data and classify test data in around 90% accuracy. The results of this paper show potentials of induction and symbolic reasoning to feature learning or pattern recognition from raw data.

Introduction

- Deep learning is powerful for *feature learning*, while it does neither show what is learned nor explain why an output is obtained.
- ILP can learn human-readable hypotheses from small amounts of data, which enables to accumulate learned results as knowledge and to share them by humans.
- The goal of this study is to realize feature learning from raw data using ILP techniques.

Methodology

- An image (in black, white or grayscale) is presented by $28 \times 28 = 784$ pixels where each pixel is an integer value from 0 to 255.
- An image is then represented as a vector $\boldsymbol{v} \in \mathbb{R}^{784}$ that contains pixel values as elements.
- Each pixel x $(0 \le x \le 255)$ is transformed to the term $f^k(z)$ with a variable z where

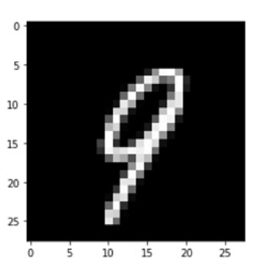
$$k = \left\lfloor \frac{x}{64} \right\rfloor + 1,$$
$$f^1(z) = f(z) \;\text{ and }\; f^{k+1}(z) = f(f^k(z)) \;\; (1 \le k \le 3).$$

The function symbol f is used to represent "closeness" of pixels. For instance, when $0 \le x_1, x_2 \le 63$, both x_1 and x_2 are represented as $f(z)$. When $x_1 = 80$ and $x_2 = 200$, x_1 is represented as $f^2(z)$ and x_2 is represented as $f^4(z)$. This representation helps to keep information of the range of shades in computing least generalization. With this encoding, a vector $\boldsymbol{v}$ is encoded into an atom having 784 arities:

$$P(t_{1.1}, \dots, t_{1.28}, t_{2.1}, \dots, t_{2.28}, \dots, t_{28.1}, \dots, t_{28.28}).$$

We omit the predicate P that is unimportant in this study. For example, the image (right) is encoded into a tuple of terms as:

$$(\underbrace{f(z), \dots, f(z)}_{28\times 6\,\text{values}}, \quad \%\text{1st to 6th rows}$$
$$\underbrace{f(z), \dots, f(z)}_{16\,\text{values}}, f^3(z), f^4(z), f^4(z), f^3(z), \underbrace{f(z), \dots, f(z)}_{8\,\text{values}}, \quad \%\text{7th row}$$
$$\underbrace{f(z), \dots, f(z)}_{14\,\text{values}}, f^2(z), f^4(z), f^4(z), f^2(z), f^2(z), f^4(z), \underbrace{f(z), \dots, f(z)}_{8\,\text{values}}, \quad \%\text{8th row} \;\; \dots\dots$$



Extracting Features

- Suppose a set of training data $C_l = \{A_1, \dots, A_n\}$ (called a *class*) where l is a label and A_i $(1 \le i \le n)$ is a first-order atom (or a tuple) representing an image.
- Compute the least generalization of C_l using an algorithm in the literature (e.g. [Nguyen& Sakama, ILP 2019]).

The obtained vector $\boldsymbol{u} \in \mathbb{R}^{784}$ is viewed as features extracted by least generalization. We call $\boldsymbol{u}$ a *feature vector* by **GEN**.

Suppose a set of training data $D_l = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$ where l is a label and $\boldsymbol{v}_k \in \mathbb{R}^{784}$ $(1 \le k \le n)$ is a vector representing an image. Put

$$\boldsymbol{v}_k = (x^k_{1.1}, \dots, x^k_{1.28}, x^k_{2.1}, \dots, x^k_{2.28}, \dots, x^k_{28.1}, \dots, x^k_{28.28}) \quad (1 \le k \le n)$$

where x^k_{ij} is a pixel value. Then, define

$$S_{ij} = \{\, x^k_{ij} \mid 1 \le k \le n \,\} \quad (1 \le i, j \le 28).$$

S_{ij} is a collection of pixel values at the location (i, j) from training data. Then, **FRQ** and **AVE** are defined as follows.

FRQ: Select the integer value u_{ij} that appears most frequently in S_{ij}.

AVE: Compute $w_{ij} = \lfloor v_{ij} \rfloor$ where v_{ij} is the average value of elements in S_{ij}.

We call $\boldsymbol{u} = (u_{ij})$ (resp. $\boldsymbol{w} = (w_{ij})$) *a feature vector* by **FRQ** (resp. **AVE**).

Classification of Images

We next use the result of extracted features for classifying unlabelled test data. When there are m classes, the classification is done using the following algorithm.

Input: a vector $\boldsymbol{v}$ representing an unlabelled 28×28 image, and the set of feature vectors of training data: $S = \{\, \boldsymbol{u}_k \mid k = 1, \dots, m \,\}$.

Output: the label of $\boldsymbol{v}$.

1. For each class k $(1 \le k \le m)$, compute
$$D_k = \sum_{ij} |\, u_{ij} - v_{ij} \,| \qquad (1 \le i, j \le 28)$$
where u_{ij} is an element in $\boldsymbol{u}_k$ and v_{ij} is an element in $\boldsymbol{v}$.
2. Return $l = k$ as the label of $\boldsymbol{v}$ where D_l is minimal among $D_1, \dots, D_m$.

The set S of feature vectors is obtained by one of **GEN**, **FRQ**, and **AVE**. The label of a testing data is determined in a way that the sum of differences between pixel values in each location is minimal.

Experimental Results

- The experiments use two datasets, MINST hand-written digits and Fashion MNIST. Each dataset is split into two parts: the training set (60,000 images) and the test set (10,000 images).
- The experimental testing is done by two stages: (i) extracting features from training data using **GEN**, **FRQ**, and **AVE**; and (ii) classifying test data and computing their *Precision*, *Recall*, and *Accuracy* to evaluate the methods.

Fig.1 shows the results of (i) and Fig.2 shows the results of (ii).

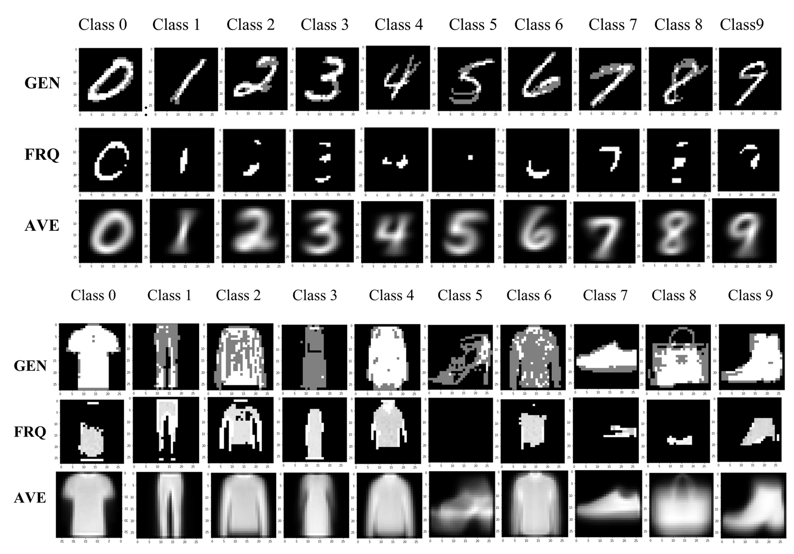


Figure 1: Features of the MNIST dataset (top) and Fashion-MNIST dataset (bottom) obtained by GEN, FRQ, and AVE

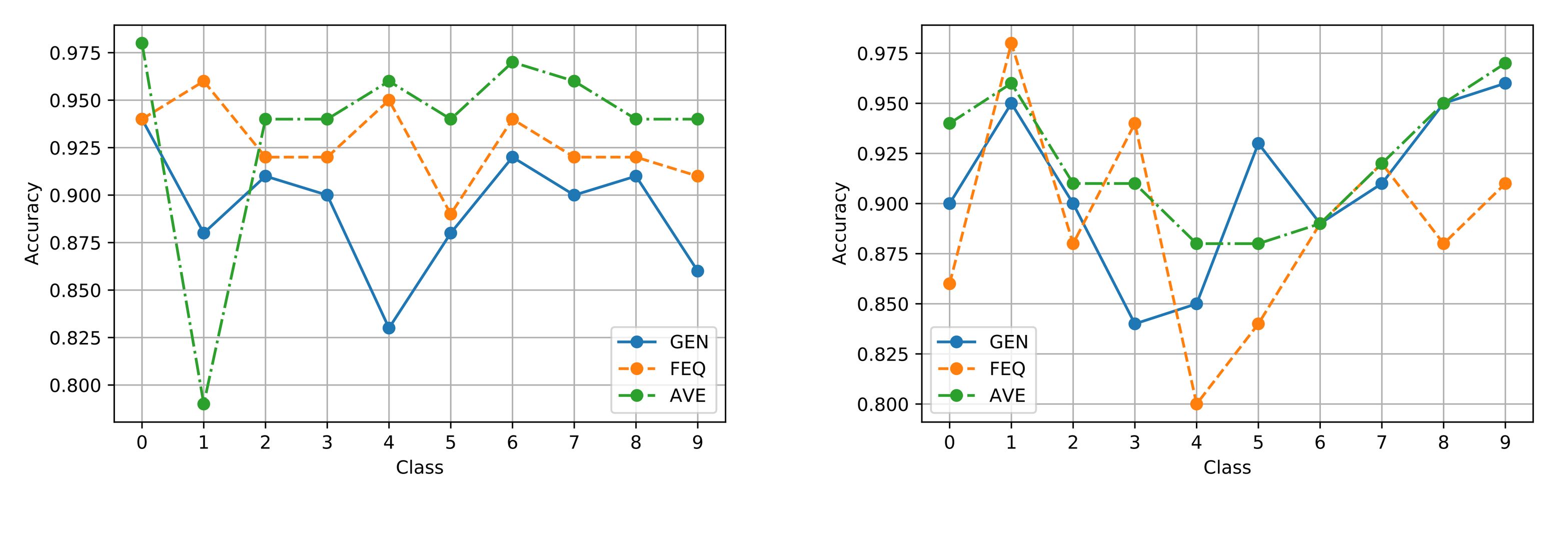


Figure 2: The accuracy on the MNIST hand-written digits dataset (left) and the Fashion-MNIST dataset (right).

Conclusion

- This study introduced new methods that learn features of labelled images by symbolic reasoning. The approach is purely symbolic and does not use NN for learning from image data.
- Experimental results show that the proposed methods successfully capture features from training data and classify test data in around 90% accuracies.
- We continue experiments to verify the effect of noise in images, and investigate robust and effective representation of raw data in terms of symbolic expressions.