산업공학특론 중간발표 #1

2015. 09. 30

Data mining center

석사과정 양호성

Fayyaz, Mohsen, et al. "Feature Representation for Online Signature Verification." *arXiv preprint arXiv:1505.08153* (2015).

Fayyaz, Mohsen, et al. "Online Signature Verification Based on Feature Representation." International Symposium on Artificial Intelligence and Signal Processing (AISP), 2015

목차

Introduction

People recognition system	Physiological biometrics	Physical part of the human body e.g. fingerprint, retina
	Behavioral biometrics	Behavioral biometrics : characteristics and behaviors of the human e.g. signature, voice
Recognition	Identification	Specify which user provides a given value among a set of known users Multi-class classification
	Verification	Determines specific known user or is a forgery One-class classification
Verification type	Off-line	Input data : X-Y coordinates
	On-line	Input data : X-Y coordinates + time, pressure, pen-up, azimuth

Related work –verification process

Preprocessing

- Smoothing, rotation, normalization

Feature Extraction

- Key steps for verification
- Non-common : DTW(Dynamic time warping), HMM, PCA

Classification

- Smoothing, rotation, normalization
- Choosing the best threshold is crucial step

Table 1 List of common features

List of common features	
#	Description
1	Coordinate $x(t)$
2	Coordinate $y(t)$
3	Pressure $p(t)$
4	Time stamp
5	Absolute position, $r(t) = \sqrt{x^2(t) + y^2(t)}$
6	Velocity in x, $v_x(t)$
7	Velocity in y, $v_y(t)$
8	Absolute velocity, $v(t) = \sqrt{v_x{}^2(t) + v_y{}^2(t)}$
9	Velocity of r(t), $v_r(t)$
10	Acceleration in x, $a_x(t)$
11	Acceleration in y, $a_y(t)$
12	Absolute acceleration, $a(t) = \sqrt{a_x{}^2(t) + a_y{}^2(t)}$

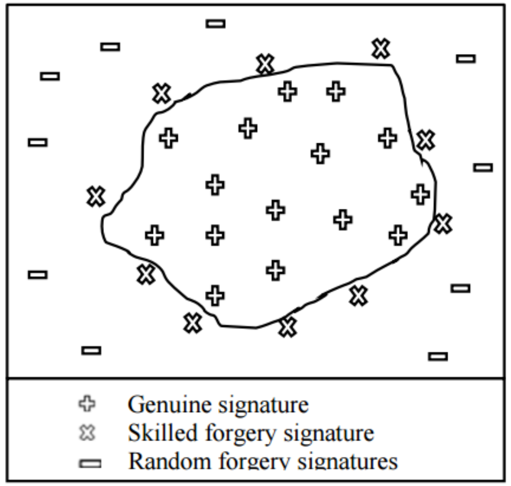


Figure 2 Example of signature model for each user

Related work - Literature

Function Based Approach

- Matching process is done using the time series data points of a signature
- Ex) DTW, HMM
- Complex and slower than Feature based

Feature Based Approach

- Matching process is done using descriptive features of a signature
- Ex) PCA
- How to derive good set of features

Methodology

Deep learning

- Learn high-level features = learning unsupervised discriminative features autonomously

Auto-encoder

- Unsupervised learning architecture used to pre-train deep networks
- Sparse autoencoder model can effectively realize feature extraction

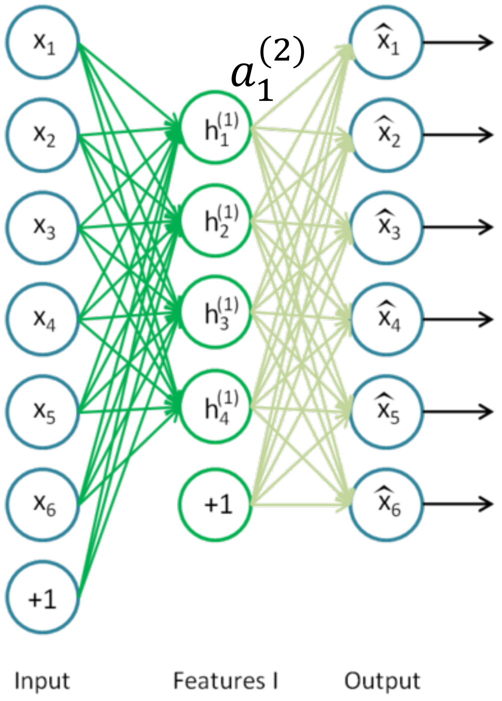
Convolution and pooling

- Locally connected Networks
- Redundant data have been neglected by picking up random patches of raw data and convolving them
- images have the “stationarity” property, which implies that features that are useful in one region are also likely to be useful for other regions.

Methodology

Auto-encoder

- Unsupervised learning architecture used to pre-train deep networks
- Sparse autoencoder model can effectively realize feature extraction



$$J_{\text{sparse}}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} \mathrm{KL}(\rho \| \hat{\rho}_j),$$

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} \left[a_j^{(2)}(x^{(i)}) \right]$$

$\hat{\rho}_j = \rho,$ ρ: sparsity parameter

$$\mathrm{KL}(\rho \| \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$$

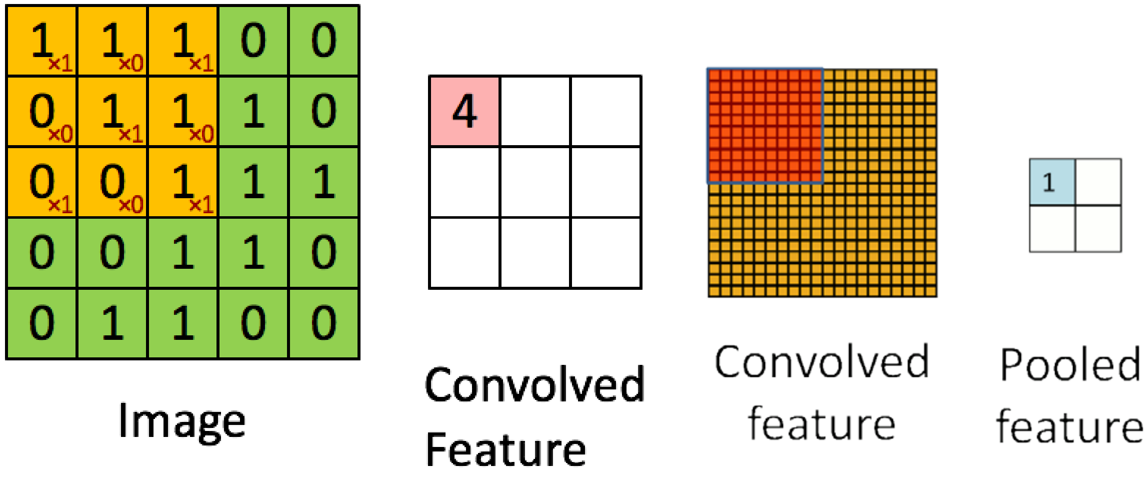
KL-divergence is a standard function for measuring how different two different distributions are.

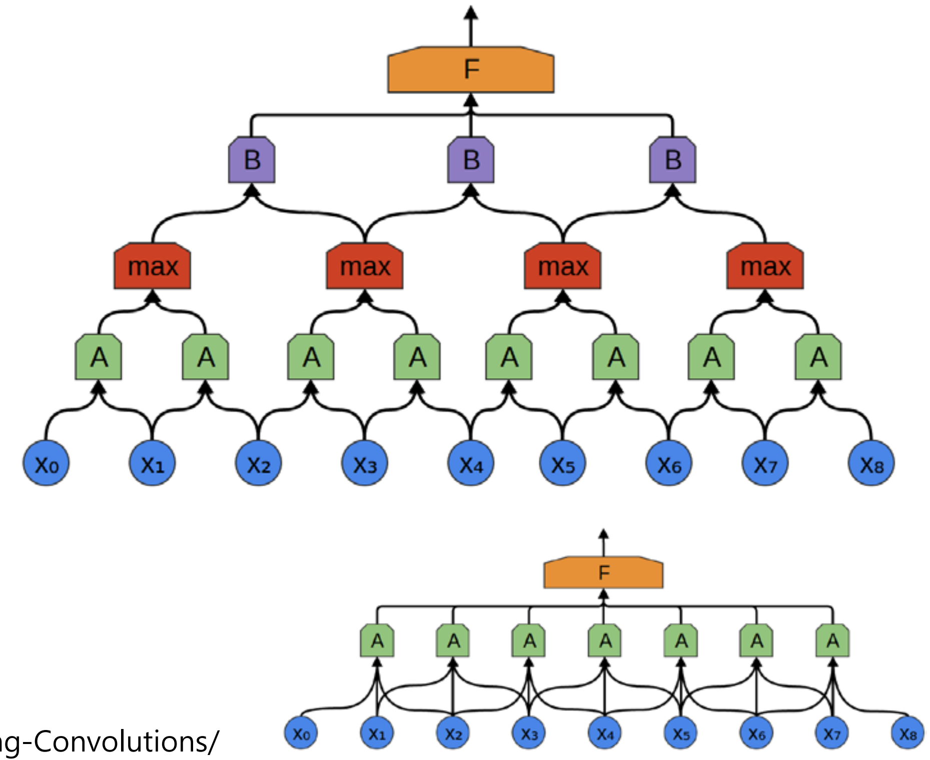
https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf

Methodology

Convolution and pooling

- Redundant data have been neglected by picking up random patches of raw data and convolving them
- images have the “stationarity” property, which implies that features that are useful in one region are also likely to be useful for other regions.





http://deeplearning.stanford.edu/tutorial/ http://colah.github.io/posts/2014-07-Understanding-Convolutions/

Proposed System

Feature Learning

- Features are learned by autoencoder

Classification

- One-class classifiers
- Neural Network, KNN, Support Vector Data Description (SVDD), Gaussian

Verification

- New unknown signatures are compared

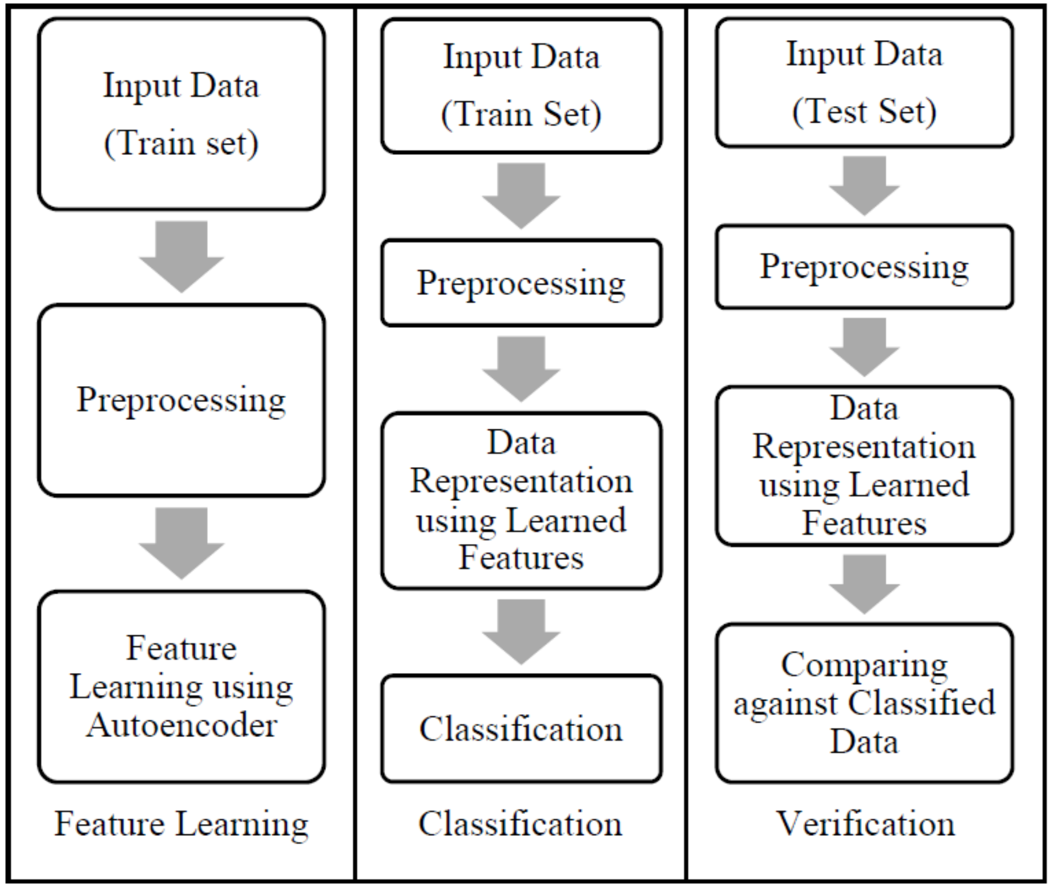


Figure 2 proposed system architecture

Three main phases

Preprocessing

- Normalizing size of the signature
- PCA, Whitening

Feature learning

- Raw data -> divide into small patches -> convolution -> mean pooling -> obtain pooled convolved features
- Using spare auto-encoder

Classification

- One-class classifier(Gaussian classifier)

Verification

- New unknown signatures are compared

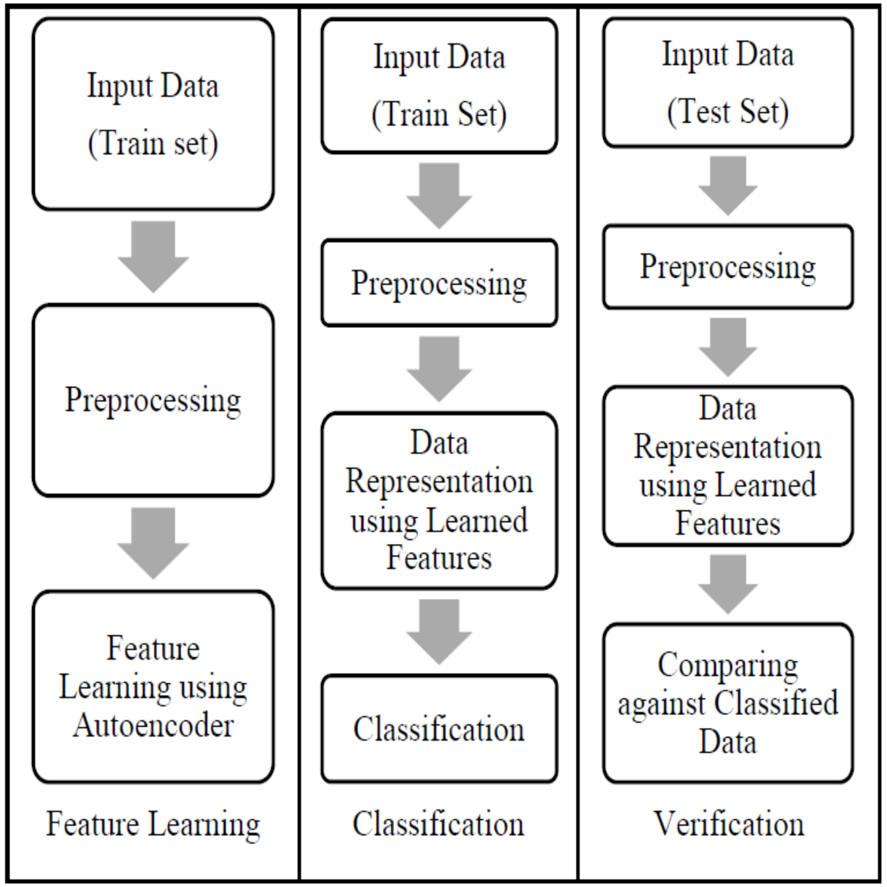


Figure 2 proposed system architecture

Experimental Results

Datasets

- SVC2004, SUSIG, ATVS

SVC2004	100 sets of sign data Each set 20 genuine, 20 skilled forgeries
SUSIG	Visual : genuine 10, forgery 10 100 users Blind : genuine 10, forgery 10 100 users
ATVS-SSig	Synthetic database 25 signs from 350 users

Evaluation

- ROC curve : x- False positive rate, y- True positive rate
- Equal error rate(EER) : rate at which both false positive and false negative are equal
- AUC curve : Higher value = better separation between target and outlier objects

		Actual	
		0	**1**
Predicted	0	True Negative	False Negative
	1	False Positive	True Positive

Experimental Results

Model description

- Hidden size : 2000
- Iteration : 700 (L-BFGS)
- Classifier : Gaussian classifier
- Feature learning dataset : ATVS (17,500 signatures)
- Classification and verification : SVC2004, SUSIG

Table 5 Different online signature verification methods for SVC2004

Method	EER (%)
Gruber, et al. [22]	6.84
Mohammadi and Faez [13]	6.33
Barkoula, et al. [9]	5.33
Yahyatabar, et al. [11]	4.58
Yeung, et al. [25]	2.89
Ansari, et al. [6]	1.65
Fayyaz, et al. [29]	2.15
Proposed Method	**0.83**

Table 3 EER Experiment results with different hidden size for SVC2004 and SUSIG

Iteration	100		200		300		400		500		600		700	
Hidden size	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG	SVC	SUSIG
500	1.7	5.02	1.65	4.94	1.60	4.77	1.60	4.74	1.55	4.57	1.45	4.07	1.03	3.23
1000	1.25	4.90	1.14	4.57	1.14	4.24	1.08	3.72	1.06	3.72	1.03	3.51	1.03	2.70
1500	1.25	3.06	1.20	2.91	1.20	2.87	1.15	2.78	1.15	2.56	1.10	2.53	1.05	2.40
2000	1.00	2.00	0.93	1.98	0.92	1.75	0.90	1.51	0.88	1.26	0.85	1.02	0.83	0.77
2500	1.05	2.56	1.00	2.52	0.90	2.39	0.89	2.36	0.80	2.32	0.77	2.20	0.73	2.15
3000	1.03	2.67	1.01	2.57	0.98	2.52	0.96	2.41	0.88	2.34	0.88	2.16	0.78	2.05

AUC

700	
SVC	SUSIG
0.993	0.988
0.994	0.990
0.995	0.991
0.996	0.995
0.996	0.992
0.996	0.992

ɔle 6 Different online signature verification methods for SUSIG

Method	EER (%)
Khalil, et al. [30]	3.06
Napa and Memon [14]	2.91
Kholmatov and Yanikoglu [26]	2.10
Ibrahim, et al. [31]	1.59
Ansari, et al. [6]	1.23
Proposed Method	**0.77**

의의 / 느낀점

1 Signature verification 문제는 오랜 시간 논의되어 온 문제

- 1980년대부터 다양한 방법으로 연구가 진행됨
- Pattern recognition 쪽에서 현재 여전히 연구가 진행

2 Deep learning 기법이 Signature dataset에 성공적으로 적용될 수 있다는 가능성을 보여줌

3 향후 signature verification 의 필요성에 대한 의문

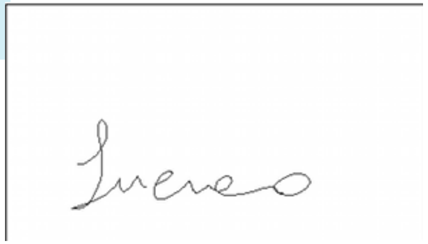
- 충분히 error 가 낮고, 앞으로 signature 이외에 더 많은 biometric verification 방법이 활용 될 것으로 예상
- **하지만** time-series & deep learning 공부하는 데는 좋은 주제인 듯함

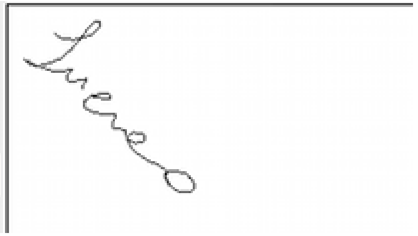
문제점 / 개선방안

1 Feature Extraction에 Synthetic data를 이용함 → Real dataset을 사용
: BiosecurID-SONOF DB 이라고 ATVS에서 최신에 공개한 132명의 real signature data가 있음

2 Raw data가 아닌 PCA를 이용해 나온 feature 를 사용 → Time-series 정보를 이용하지 못함

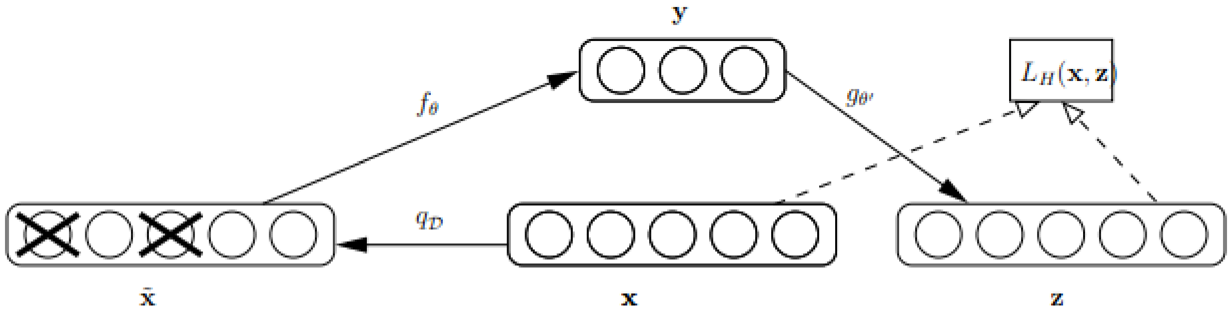
3 Data가 너무 잘 정돈된 상황에서 얻음(Noise가 적음) → 문제를 더 어렵게 만들어 보자 : Noise 추가(rotation 등)
다른 방법론을 적용해 보자 : RNN/LSTM, **Denoising autoencoder**





Denoising autoencoder

- Simple idea : destroying information of randomly selected input features; train to restore it.
- 0-masking noise (now called «dropout» noise)
- Will encourage representation that is robust to small perturbations of the input



Dataset	SVM$_{rbf}$	SVM$_{poly}$	DBN-1	SAA-3	DBN-3	SdA-3 (ν)
basic	**3.03±0.15**	3.69±0.17	3.94±0.17	3.46±0.16	3.11±0.15	**2.80±0.14** (10%)
rot	11.11±0.28	15.42±0.32	14.69±0.31	**10.30±0.27**	**10.30±0.27**	**10.29±0.27** (10%)
bg-rand	14.58±0.31	16.62±0.33	9.80±0.26	11.28±0.28	**6.73±0.22**	10.38±0.27 (40%)
bg-img	22.61±0.37	24.01±0.37	**16.15±0.32**	23.00±0.37	**16.31±0.32**	**16.68±0.33** (25%)
rot-bg-img	55.18±0.44	56.41±0.43	52.21±0.44	51.93±0.44	47.39±0.44	**44.49±0.44** (25%)

MNIST + Noise
- Rotation
- Background random pixel
- Background add img

Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.

추후 일정

예상 시나리오

Noise가 추가된 데이터에 대해서 LSTM, Denoising Autoencoder 등의 방법을 사용하여 이 논문의 방법보다 robust한 feature를 뽑아서 정확도를 향상시킬 수 있었다

10□	11□	12□

이 논문의 방법론 및 LSTM 으로 구현해서 성능 비교

문제를 더 어렵게 만들어서(rotation 추가 등)
Denoising Autoencoder 등의 방법론으로 실험

추가 개선 방향 idea 에 대한 실험

논문 작성

감사합니다