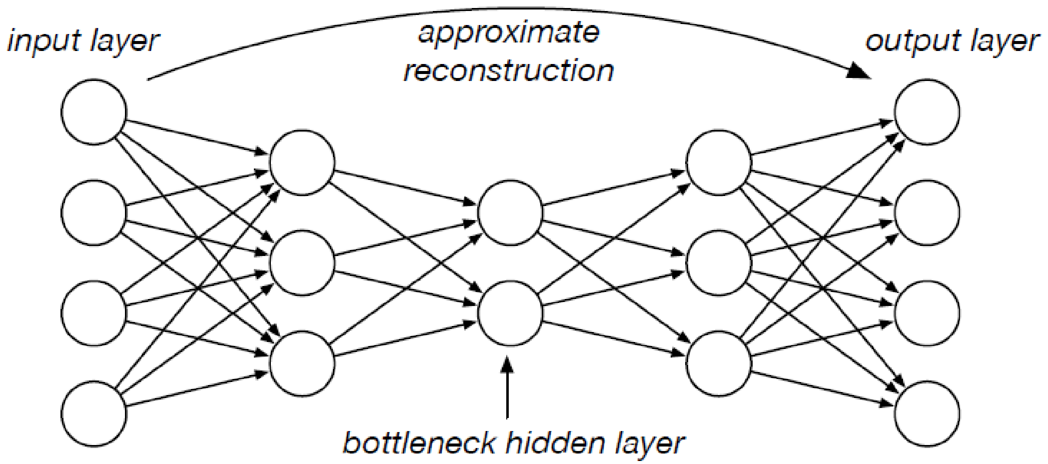
DAGMM :
DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL FOR UNSUPERVISED ANOMALY DETECTION

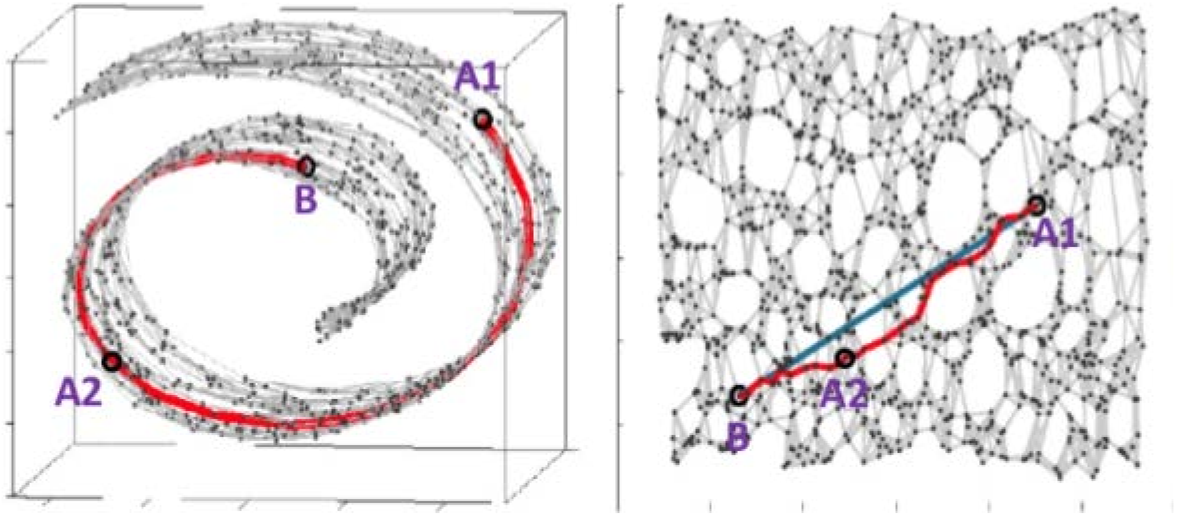
(ICLR 2018)

송민수

AutoEncoder

- Unsupervised learning
- Nonlinear dimensionality reduction
- Feature extraction
- Manifold learning





https://www.slideshare.net/NaverEngineering/ss-96581209

- $min\ (x_i - \hat{x}_i)^2$
- input=output

Introduction

- DAGMM 모델
  - AE를 통해 low-dim representation과 reconstruction error를 만들고 이 값들을 GMM의 input으로 넣는 모델
  - GMM의 parameter를 EM-algorithm으로 iterative하게 구하는 것이 아니라 NN을 통해 구한다.

Introduction

- density estimation은 이상치 탐지 방법론 중 하나
  - Probability가 낮은 area에 위치한 값을 anomaly라고 판단

- 그런데 차원이 높은 경우 density estimation에 어려움이 존재
  - 이를 위해 차원을 축소하여 Low-dimensional latent space를 만든 뒤
  - Density estimation을 하고자 하는 것

- 이 때 차원축소는 AE로 하고 density estimation은 GMM을 통해 진행

Introduction

- 해당 paper에서 propose하는 내용
  - 차원을 축소하면서 정보손실 최소화
  - DAGMM이 그냥 GMM보다 더 복잡한 구조를 잡을 수 있다
  - AE가 local optima에 갇히기 쉬운데 우리는 그렇지 않다

DAGMM

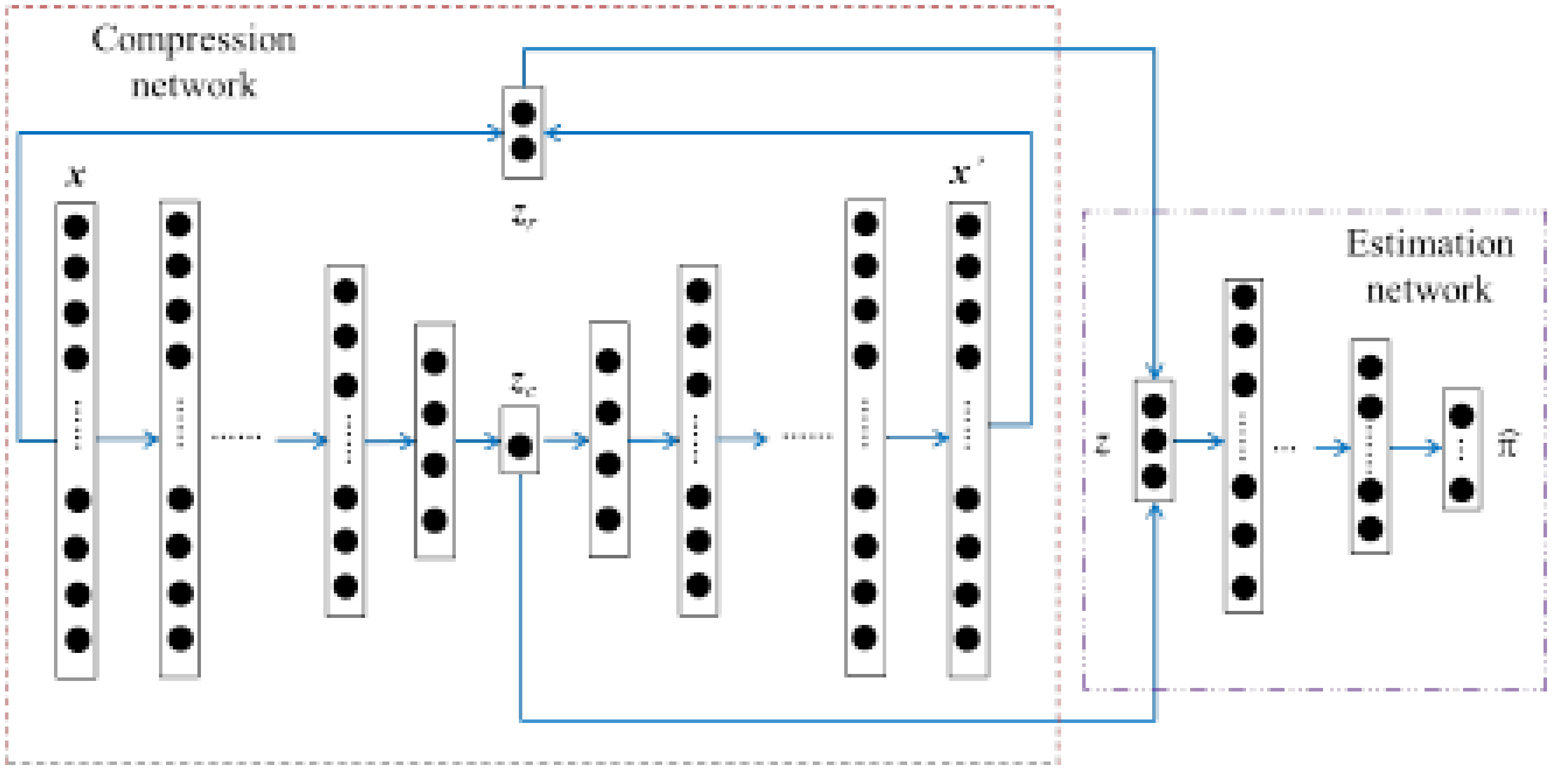


Figure 2: An overview on Deep Autoencoding Gaussian Mixture Model

- 크게 compression network + estimation network 로 이루어짐
- 위의 그림은 multi dimension의 data 1개

DAGMM : compression network

- Compression network에서는 low-dim latent vector를 만든다
  - 1) AE를 통해 줄어든 dim vector
  - 2) reconstruction error를 통해 얻은 vector

- 위의 두가지 vector를 concat하여서 Estimation network의 input으로 사용
- 해당하는 vector는 두가지 정보를 갖고 있는 것

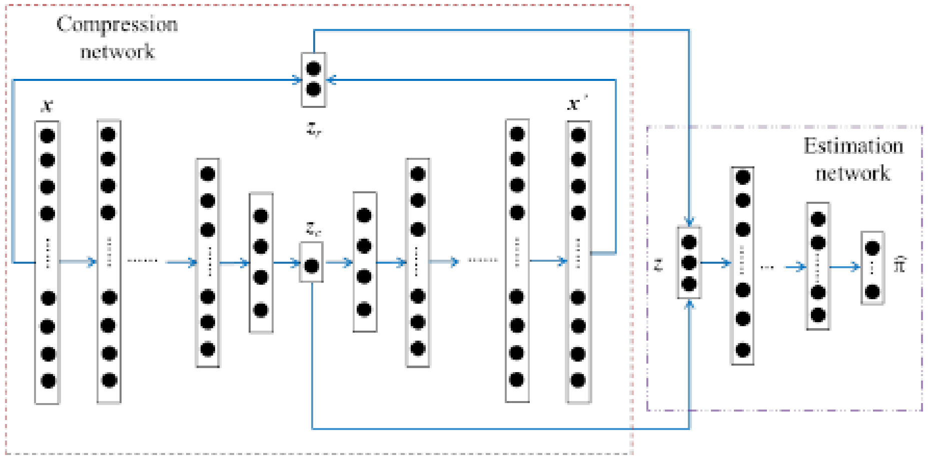


Figure 2: An overview on Deep Autoencoding Gaussian Mixture Model

DAGMM : estimation network

- Compression network에서 만든 low-dim latent vector를 input으로 받음
- Performs density estimation under the framework of GMM

- EM을 이용하지 않고 NN으로 각 sample마다 membership을 예측하도록 훈련
- NN도 EM처럼 반복적으로 parameter를 update하기에 GMM에 이용 가능

- output은 K-dimensional vector (K개의 그룹이 있다고 가정하면)

$$\vec{p} = NN(\vec{z}; \theta_m), \ \ \hat{\gamma} = softmax(\vec{p})$$

DAGMM : estimation network

- Parameter
  - ϕ : unknown mixture-component distribution
  - μ : mixture mean
  - Σ : mixture covariance

- NN을 훈련시키는 과정을 통해 아래와 같이 parameter estimation
- Where $\hat{\gamma}_i$: membership prediction for i

$$\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}, \qquad \hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} \mathbf{z}_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}, \qquad \hat{\boldsymbol{\Sigma}}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} (\mathbf{z}_i - \hat{\mu}_k)(\mathbf{z}_i - \hat{\mu}_k)^T}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}.$$

DAGMM : estimation network

- With estimated parameters, sample energy can be further inferred by

$$E(\mathbf{z}) = -\log\left(\sum_{k=1}^{K}\hat{\phi}_k \frac{\exp\left(-\frac{1}{2}(\mathbf{z}-\hat{\mu}_k)^T\hat{\Sigma}_k^{-1}(\mathbf{z}-\hat{\mu}_k)\right)}{\sqrt{|2\pi\hat{\Sigma}_k|}}\right).$$

- Test할 때, 위의 값이 높게 나오면 anomaly!
- Mixture 확률분포에 –log를 취한 것

DAGMM : objective function

- 마지막식은 cov의 singularity를 방지하기 위한 penalty term
  - Diagonal term이 너무 작아지지 않도록

$$J(\theta_e, \theta_d, \theta_m) = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x}_i, \mathbf{x}_i') + \frac{\lambda_1}{N}\sum_{i=1}^{N} E(\mathbf{z}_i) + \lambda_2 P(\hat{\mathbf{\Sigma}}).$$

Conclusion

- Dimension reduction and density estimation by end-to-end
- Estimation network가 compression network에 대해서 regularization 역할도 해준다.

My Conclusion

- 딥러닝 모델을 이용하여 GMM을 했다는 아이디어가 참신했다.
- 기본적인 AE가 아닌 다른 모델을 이용하여 dimension reduction을 하면 어떨까?
- Normal과 Abnormal 차이를 더 극대화할 수 있는 방법은?
- GMM을 더 잘하기 위한 방법은?