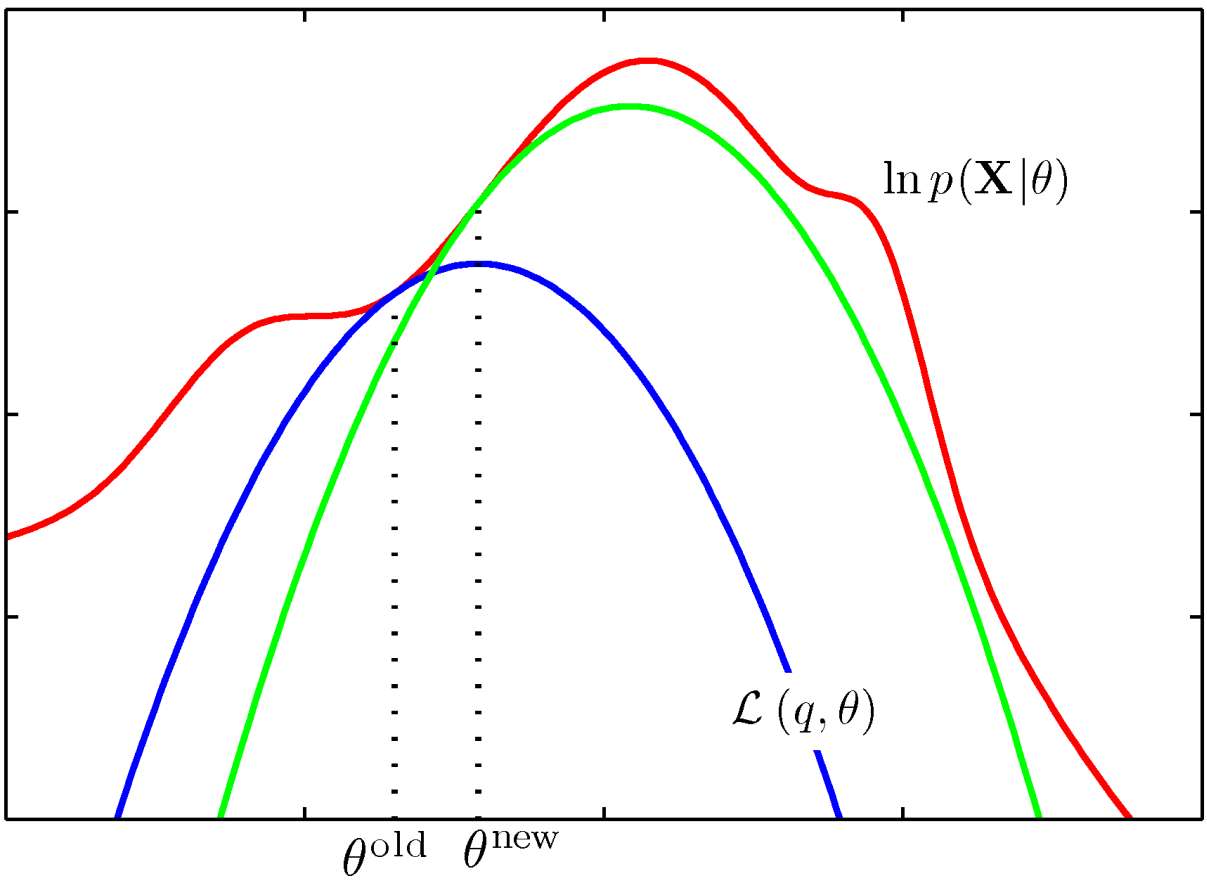
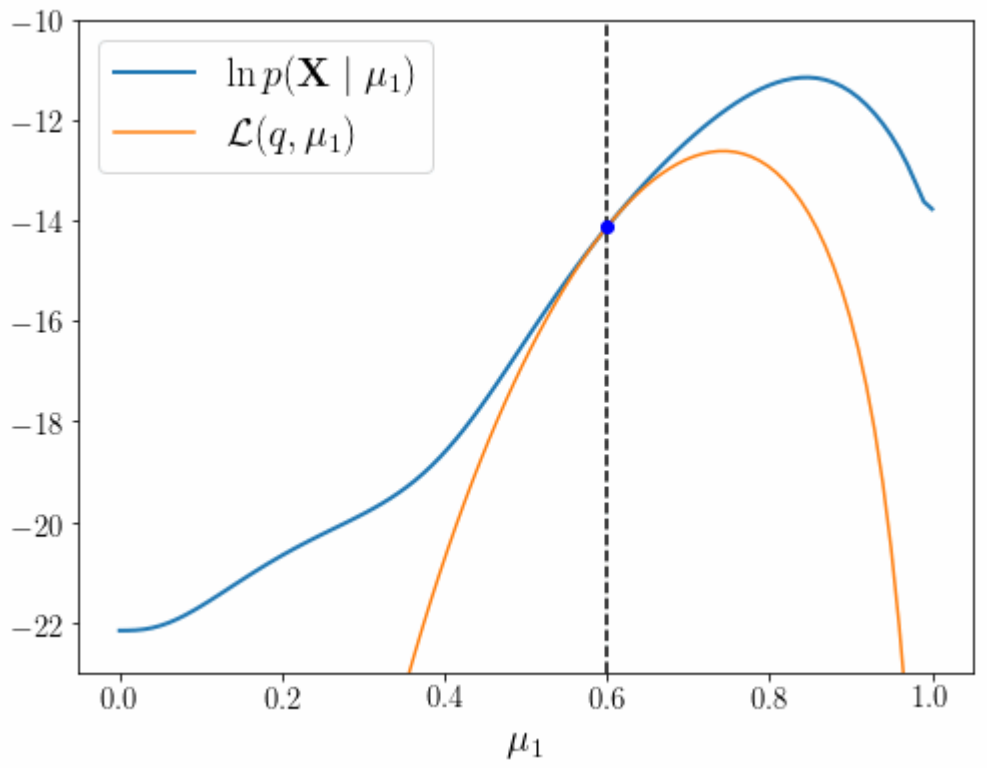
Expectation Maximization (EM)

- EM 알고리즘의 기본적인 아이디어
    - 이미 알려지지 않은 변수를 추가하여 추가 변수에 대한 함수를 최대화 하는 것
    - **새로운 Latent 변수를 도입한 우도 함수에서 Expectation Step을 통해 다루기 쉬운 Lower Bound를 구하고 이를 최대화 하는 과정인 Maximization Step을 반복하여 최대 우도 추정량을 추정**해내는 알고리즘
      - unsupervised Learning에서 주로 사용되는 알고리즘, Clustering에 사용
- **Expectation Step**
    - 주어진 임의의 파라미터 초기값에서 Likelihood와 최대한 근사한 likelihood 값을 계산
- **Maximization Step**
    - E-step에서 계산된 likelihood를 최대화(maximize)하는 새로운 파라미터값을 얻음
- 

    - 선 **임의의 파라미터** θ^{old} 로부터 EM 알고리즘이 시작됨
    - 해당 지점에서 log likelihood 함수와 최대한 근사한 L 함수를 만들게 된다.
        - 이 때 L 함수는 θ 에 대해 concave 함수(위로 볼록한 함수 <-> convex)이으로 위와 같은 그림이 된다. (파란색)
        - 이 단계가 E-Step에서 이루어진다.
    - 얻어진 L 함수를 최대화하는 새로운 파라미터 값을 선정하게 된다.

        - 위의 그림에서는 θ^{new} 이다.
        - 이를 이용하여 새로운 L 함수를 얻음. (초록색)
        - 이 단계가 M-Step 단계이다.
    - 수렴 조건을 만족할 때까지 반복하게 된다.



- 장점
    - 각 Step 마다 우도 함수를 증가 시키는 추정치를 계산할 수 있기 때문에 계산의 안정성이 보장
    - 원래의 우도 함수를 최대화 하는 문제보다 EM 알고리즘을 적용하는 것이 더 쉬움
    - Maximization Step의 해는 Closed Form으로 나타나는 경우가 많으므로 계산이 훨씬 쉬워짐
- 단점
    - Iterative 방법들의 단점은 Local Maxima에 빠질 위험이 있음
    - 수렴 속도가 일반적으로 느림
- 사용 분야
    - GMM, 결측치 추정, 인자 분석, Hidden Markov 모델 등에서 사용

Neural Expectation Maximization (NIPS 2017)

Introduction

- unsupervised learning의 중요한 목표 중 하나는 useful representation을 학습하는 것
- rotation과 lighting과 같은 feature을 분리하는 representation을 생성하는 등 이러한 방법은 주로 single object case에서만 연구되었지만, 실제 물리적 상호작용과 같은 작업에서는 multiple object와 그 관계를 식별해야하는 경우가 많지만, 현재 연구들에서는 multiple distributed 및 disentangled representation을 생성하는 것은 어려운 문제이며, 이는 신경 과학 분야에서 계속 연구가 진행되고 있음

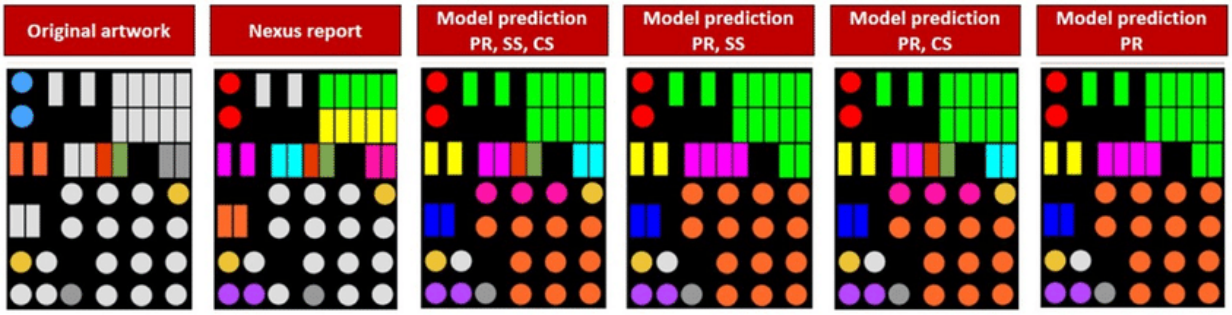


Fig1. The original art work, the nexus report, and the model predicted grouping. We used K-NN model with all the three features here

-
- 이러한 문제를 해결하기 위해서 각 객체에 대한 **separate representation을 학습**하는 것이 필요
- 즉, **perceptual grouping** 이 필요
- 이를 위해 데이터의 통계적 구조에 기반하여 unsupervised 방식으로 개별 객체들을 그룹핑하고 효율적으로 representation 하는 방법을 학습
- object representation θ_k이 주어진 이미지 통계 모델인 분포 군 $P(x|\theta_k)$에 접근할 수 있다면, 이러한 분포의 mixture에서 추론하여 클러스터링을 수행
- EM을 사용하여 mixture(θ_1, ..., θ_k)의 매개변수에 대한 Maximum Likelihood Estimate(MLE)를 계산함으로써 각 객체에 대한 픽셀의 클러스터링과 representation을 얻을 수 있음
- 실제로 $P(x|\theta_k)$를 알 수 없기 때문에 Neural EM은 Neural Network를 통해 mixture를 매개변수화 하고 EM Step의 반복을 통해 backpropagation 하여 학습을 수행

Problem statement and main contribution

- 신경 과학 관점에서 visual mechanism을 연구
- 모양은 서로 다르지만 겹치는 여러 객체들을 인식하고 추적하는 방법?
- ML 관점에서 이러한 문제는 **pixel-wise clustering** 문제
- EM 알고리즘은 RNN의 프레임워크에서 구현될 수 있으며, unsupervised learning에 적용 가능

기존 EM 알고리즘

- 각 이미지의 x $\in \mathbb{R}^D$ 는 벡터 θ_1 , ... , $\theta_K \in \mathbb{R}^M$ 으로 매개변수화 된 component K의 spatial mixture로 모델링
- 미분 가능한 non-linear function f_ϕ (a neural network with weight ϕ) 은 이러한 representation $\{\theta_k\}_{k=1}^K$ 을 별도의 pixel-wise distribution을 위한 매개변수 $\psi_{i,k}$ = $f_\phi(\theta_k)_i$로 변환하는데 사용
- 이러한 distribution은 일반적으로 Bernoulli 또는 Gaussian이며, 여기서 $\phi_{i,k}$는 single probability or mean
- binary latent variables $Z = [z_{i,k}] \in [0,1]^{D\times K}$ 는 unknown true pixel 할당을 인코딩
- pixel i가 component k 에 의해 생성되었다면 $z_{i,k} = 1$ 이고, $\sum_k z_{i,k} = 1$
  - e.g. 0.45 + 0.55 = 1, known true pixel이라면 1 false pixel은 0
- $\theta = [\theta_1, \ldots, \theta_k]$ 가 주어졌을 때 x에 대한 likelihood는 아래와 같음
  - 여기서 $\pi = (\pi_1, \ldots, \pi_k)$는 mixing coefficient(or prior for z)

$$P(\boldsymbol{x}|\boldsymbol{\theta}) = \prod_{i=1}^{D} \sum_{\boldsymbol{z}_i} P(x_i, \boldsymbol{z}_i|\boldsymbol{\psi}_i) = \prod_{i=1}^{D} \sum_{k=1}^{K} \underbrace{P(z_{i,k}=1)}_{\pi_k} P(x_i|\psi_{i,k}, z_{i,k}=1).$$

-

Neural EM algorithm

- 개별 데이터를 클러스터링 하고 학습하기 위해 Neural Network와 EM 알고리즘을 결합하는 새로운 프레임워크를 제안
- primitive object representations 로 구성된 visual scene representation을 학습하는 미분 가능한 클러스터링 방법
- **Neural EM의 목표는 동일한 객체에 속하는 입력 픽셀을 그룹화 하고 (perceptual grouping) 각 객체에 대한 distributed representation θ_k 에서 정보를 효율적으로 처리하는 것**
- θ에 대한 $logP(x|\phi)$를 직접 최적화 하는 것은 z에 대한 marginalization으로 인해 어려운 반면, 많은 분포에서 $logP(x,z|\phi)$를 최적화하는 것은 훨씬 쉬움
- 따라서 EM은 이를 활용하여 expected log likelihood에 대한 lower bound를 최적화 함

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{z}} P(\mathbf{z}|\boldsymbol{x}, \boldsymbol{\psi}^{\text{old}}) \log P(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\psi}).$$

-
- 이러한 bound의 최적화는 2가지 단계를 번갈아 수행
  - E-step : 각 픽셀에 대해 previous iteration에서 주어진 θ_{old}에서 posterior probability distribution의 새로운 추정치를 계산하여 components(clusters)에 대한 픽셀의 새로운 soft assignment를 산출

$$\gamma_{i,k} := P(z_{i,k} = 1|x_i, \psi_i^{\text{old}}).$$

  -
  - M-step : E-step에서 계산된 posterior를 사용하여 예상되는 log-likelihood를 최대화 하는 θ의 구성을 찾는 것을 목표로 함

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{z}} P(\mathbf{z}|\boldsymbol{x}, \boldsymbol{\psi}^{\text{old}}) \log P(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\psi}).$$

    -
    - 본 논문에서 $P(x_i|z_{i,k} = 1, \psi_{i,k})$ = $N(x_i; \mu = \phi_{i,k}, \sigma^2)$
    - non-linear f_ϕ 로 인해 $argmax_\theta$ $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$ 에 대한 해가 존재하지 않으나, f_ϕ는 미분 가능하기 때문에 gradient ascent step를 수행하여 $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$ 를 개선 할 수 있음

$$\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^{\text{old}} + \eta \frac{\partial \mathcal{Q}}{\partial \boldsymbol{\theta}} \qquad \text{where} \qquad \frac{\partial \mathcal{Q}}{\partial \boldsymbol{\theta}_k} \propto \sum_{i=1}^{D} \gamma_{i,k}(\psi_{i,k} - x_i) \frac{\partial \psi_{i,k}}{\partial \boldsymbol{\theta}_k}.$$

    -

Unrolling Neural EM algorithm

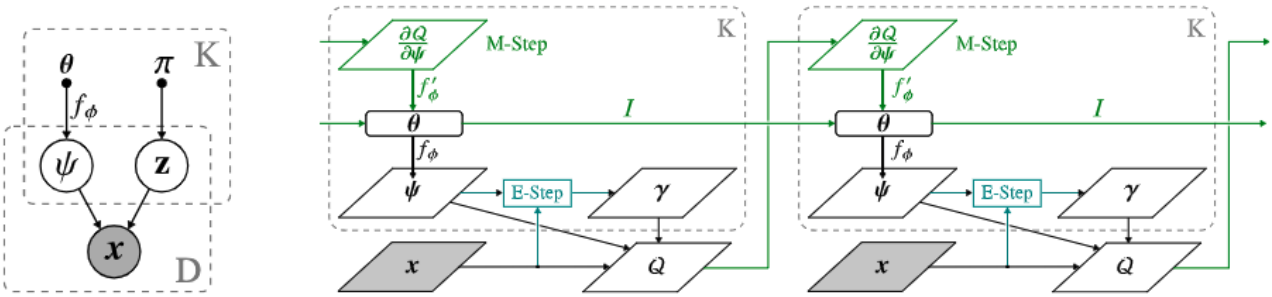


Figure 1: **left:** The probabilistic graphical model that underlies N-EM. **right:** Illustration of the computations for two steps of N-EM.

- f_ϕ 에 의해 구현된 통계 모델을 기반으로 end-to-end에서 미분 가능한 clustering procedure을 얻음
- latent variables $[\theta_1, \ldots, \theta_K]$ 는 hidden stage
- $\frac{\partial \psi}{\partial \theta}$ 는 hidden stage variable을 업데이트 하기 위한 encoder
- f_ϕ는 할당 결과 $\psi_{i,k}$ 를 생성하는 decoder
- EM은 K encoder의 ensemble에 의해 이뤄지고, decoder는 최적화를 수행

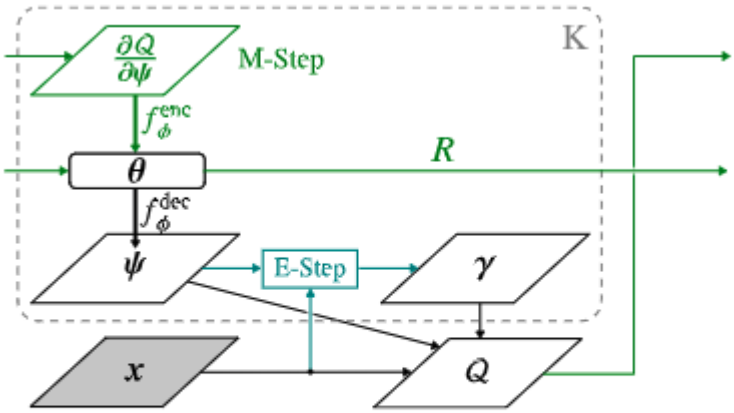


Figure 2: RNN-EM Illustration. Note the changed encoder and recurrence compared to Figure 1.

- 본 논문에서는 encoder를 신경망 구조로 대체하여 설계를 더욱 완화시키고 RNN-EM 알고리즘을 제안
- 여기서 $f_\phi^{enc}(\cdot)$, $f_\phi^{dec}(\cdot)$, $R(\cdot)$ 는 신경망 매개변수

Learning for clustering

- unsupervised clustering을 위해 two term loss function으로 f_ϕ를 학습하여 θ 에서 대상 객체에 대한 픽셀별 분포에 해당하는 매개변수 ψ로 매핑
- $$L(\boldsymbol{x}) = -\sum_{i=1}^{D}\sum_{k=1}^{K} \underbrace{\gamma_{i,k} \log P(x_i, z_{i,k}|\psi_{i,k})}_{\text{intra-cluster loss}} - \underbrace{(1-\gamma_{i,k}) D_{KL}[P(x_i)||P(x_i|\psi_{i,k}, z_{i,k})]}_{\text{inter-cluster loss}}.$$
  - 첫번째 항은 expected log-likelihood 와 같음

- 두번째 항은 out-of-cluster pixel $\gamma_{i,k} = 0$ 의 손실이 vanishing되는 문제를 방지
  - out-of-cluster prediction과 pixelwise prior 사이의 KL 발산에 불이익을 주는 inter-cluster loss를 추가
- $\gamma_{i,k} = 0$ 일 때, 이 항은 할당되지 않은 픽셀 x_i에 관한 정보를 포함하지 않도록 각 θ_k에게 정보를 줌
  - $P(x_i|z_{i,k} = 1, \psi_{i,k}) = P(x_i)$

Experimental results : non-sequential case

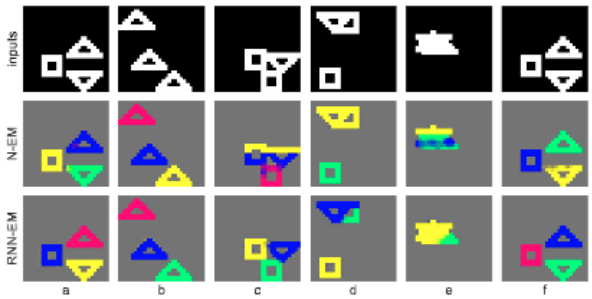


Figure 3: Groupings by RNN-EM (bottom row), N-EM (middle row) for six input images (top row). Both methods recover the individual shapes accurately when they are separated (a, b, f), even when confronted with the same shape (b). RNN-EM is able to handle most occlusion (c, d) but sometimes fails (e). The exact assignments are permutation invariant and depend on γ initialization; compare (a) and (f).

- $x = [x_i]$: 28 x 28 binary image (with 3 shapes)
- θ_k : 250-dimensional latent vector for $k = 1, \ldots, 3$
- $\psi_k = [\psi_{i,k}]$: k-th cluster의 픽셀 별 Bernoulli distributeion 의 평균 값
- $f_\phi^{dec}(\theta_k) = \psi_k$: 각 픽셀에 대한 sigmoid output $\psi_{i,k}$ 가 있는 single layer fully connected neural network
- $f_\phi^{enc}(\cdot)$: N-EM에 대한 backpropagation을 통해 얻은 gradient, RNN-EM에 대한 neural network
- $R(\cdot)$: RNN으로 이동하기 전에 각 θ_k를 (0,1) range로 변경하는 sigmoid 함수

- N-EM과 RNN-EM 방식 모두 동일한 shape을 가지더라도 분리되는 양상을 볼 수 있음
- 이미지에 occlusion이 포함된 경우 N-EM의 성능이 저하되고, RNN-EM이 일반적으로 더 안정적이고, 더 나은 그룹을 생성함

Experimental results : sequential case - Flying Shapes

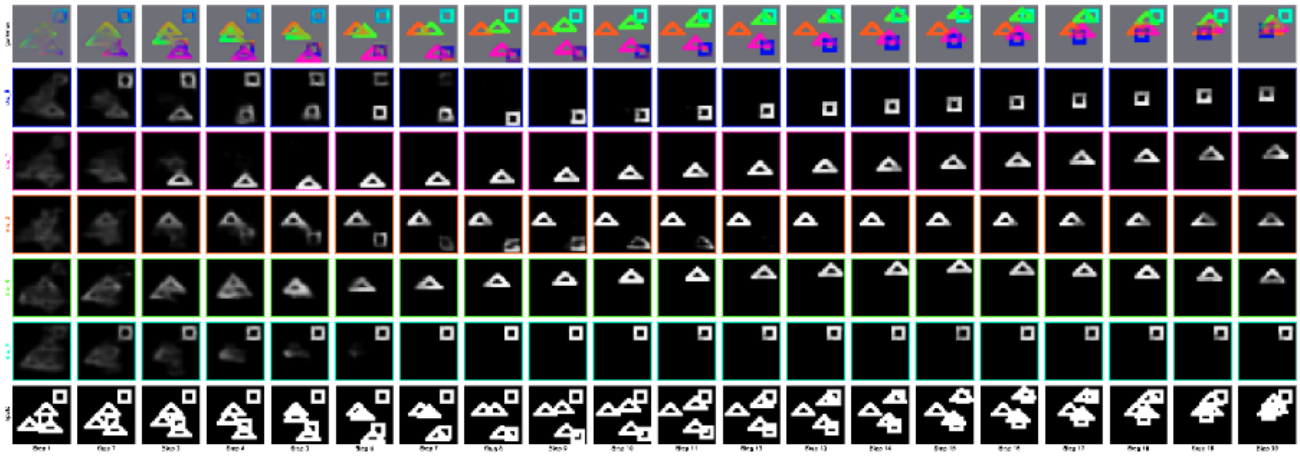


Figure 4: A sequence of 5 shapes flying along random trajectories (bottom row). The next-step prediction of each copy of the network (rows 2 to 5) and the soft-assignment of the pixels to each of the copies (top row). Observe that the network learns to separate the individual shapes as a means to efficiently solve next-step prediction. Even when many of the shapes are overlapping, as can be seen in time-steps 18-20, the network is still able to disentangle the individual shapes from the clutter.

- shape이 임의의 궤적을 따라 이동하고, 정적인 shape들이 순차적으로 확장되는 시퀀스
- 20 RNN-EM iterations, 맨 아래 행이 input, 맨 위 행이 gammas
- f_ϕ^{dec} : infoGAN의 구조에 의해 영감을 받은 convolutional layer를 포함
- θ_k : 100-dimensional sigmoidal hidden state
- 각 timestep t 에서 network는 input으로 $\gamma_k \odot (\psi_k^{t-1} - \hat{x}^t)$를 받음
  - 이 때 $\hat{x}^t$은 bitfilp noise (p=0.2)에 의해 생성된 current frame
- next-step prediction 목표는 x^t 를 x^{t+1} 로 대체하여 구현됨

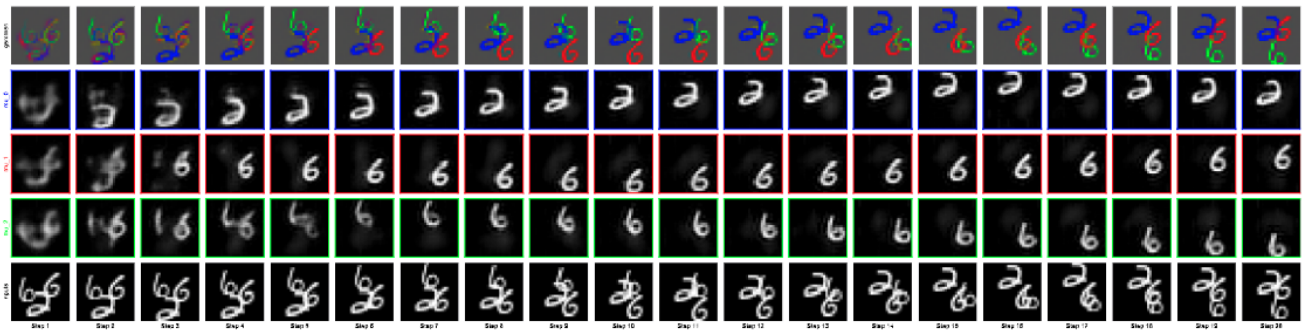


Figure 7: A sequence of 3 MNIST digits flying across random trajectories in the image (bottom row). The next-step prediction of each copy of the network (rows 2 to 4) and the soft-assignment of the pixels to each of the copies (top row). Although the network was trained (stage-wise) on sequences with two digits, it is accurately able to separate three digits.

- 위 예제와 거의 동일

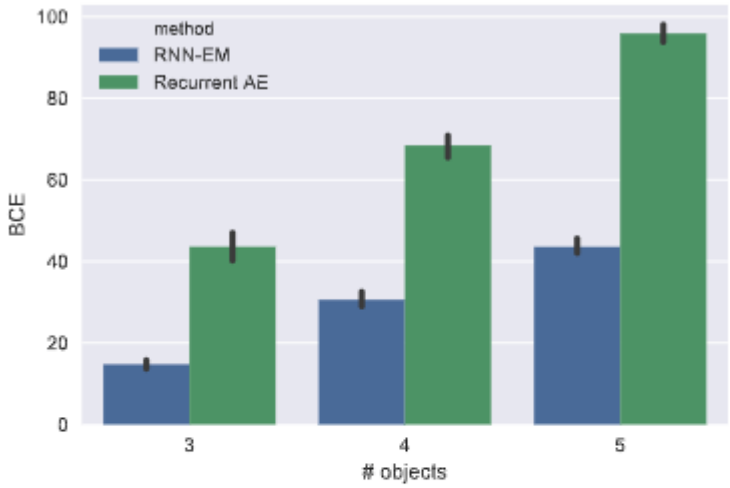


Figure 5: Binomial Cross Entropy Error obtained by RNN-EM and a recurrent autoencoder (RNN-EM with $K = 1$) on the denoising and next-step prediction task. RNN-EM produces significantly lower BCE across different numbers of objects.

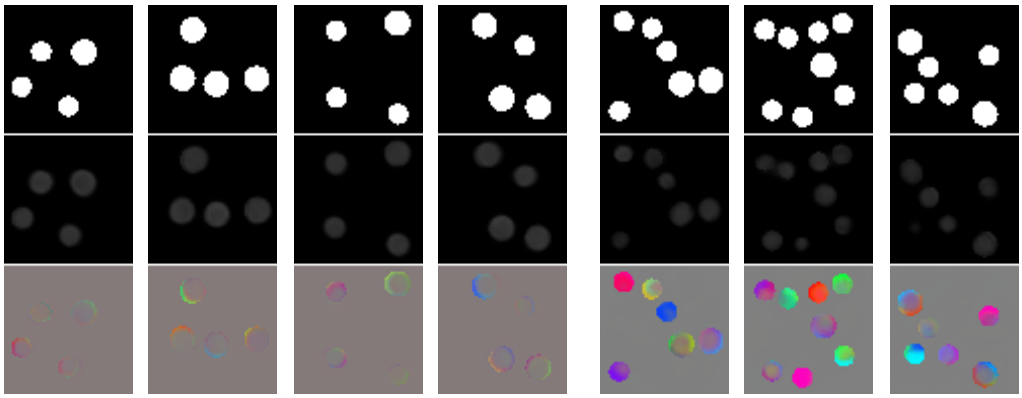
- 실험결과를 통해 RNN-EM은 객체 수가 증가할 때 훨씬 더 낮은 오류를 생성하는 것을 알 수 있음

conclusion

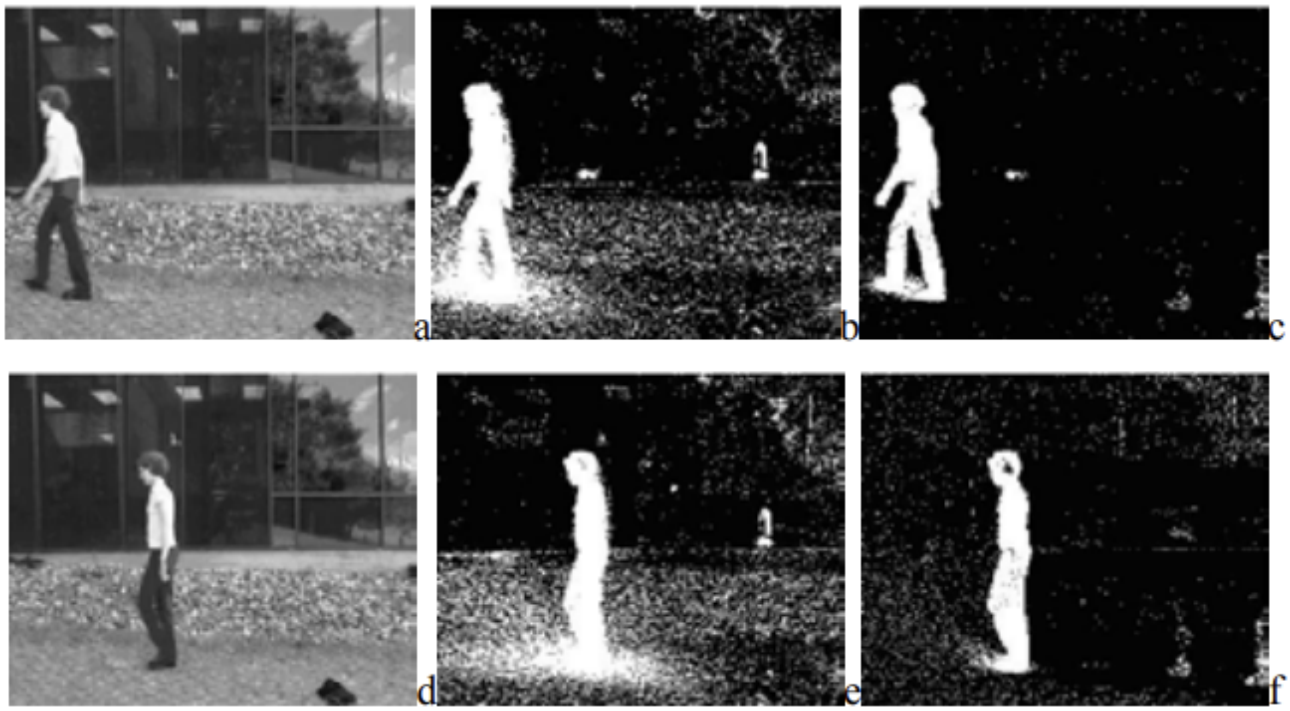
- Pros
  - unsupervised learning에 대한 RNN 모델의 흥미로운 확장판
  - EM 알고리즘과 RNN 구조를 결합하여 EM 알고리즘의 유연성을 높임
- Cons
  - 모델 구현 시 많은 트릭들이 필요함. 다른 데이터에 적용 가능한지 추가 실험 필요
  - inter-cluster loss를 설계하여 out-of-cluster pixel을 정규화 함
  - inter-cluster loss의 가중치를 주의하여 설계해야함
  - RNN-EM 모델을 학습할 때 이미지에 노이즈를 추가해야하며, 서로 다른 데이터에 대해 서로 다른 노이즈 전략이 필요함

Further ...

Relational Neural Expectation-Maximization
https://github.com/sjoerdvansteenkiste/Relational-NEM



- GMM을 이용한 background subtraction
  - Gaussian Mixutre Model : EM 알고리즘을 적용한 여러가지 모델 중 하나
  - https://www.sciencedirect.com/science/article/pii/S1877705811065258



참고자료

- https://angeloyeo.github.io/2021/02/08/GMM_and_EM.html
- https://zephyrus1111.tistory.com/89
- http://norman3.github.io/prml/docs/chapter09/4.html
- https://techblog-history-younghunjo1.tistory.com/88
- http://people.ee.duke.edu/~lcarin/Dixin9.28.2018.pdf