Neural Expectation Maximization

IDSIA — Scuola universitaria professionale della Svizzera italiana SUPSI — Università della Svizzera italiana

Sjoerd van Steenkiste*
sjoerd@idsia.ch

Klaus Greff*
klaus@idsia.ch

Jürgen Schmidhuber
juergen@idsia.ch

Summary

We introduce Neural Expectation Maximization (N-EM), a novel unsupervised framework for representation learning that splits images into distinct objects (perceptual grouping) and represent each one separately.

- Every image is modeled as a spatial mixture model with K components, each summarized by a distributed representation $\boldsymbol{\theta}_k$
- A neural network implement the statistical model of the individual components by transforming the representations $\boldsymbol{\theta}_k$ into distributions over pixel values.
- We use generalized EM to jointly infer 1) the assignment of pixels to components 2) the representations for all components.
- The result is a differentiable clustering procedure that can be trained to recover the constituent objects of a given input.
- We apply our framework to synthetic perceptual grouping tasks and empirically verify that it yields the intended behavior.
- This approach naturally extends to other domains.

Motivation

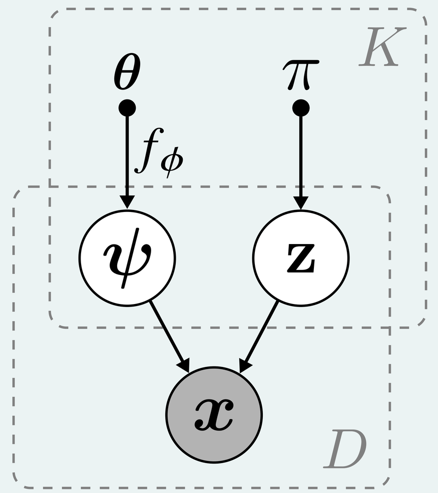
- Many high-level real world tasks such as reasoning and physical interaction require identification and manipulation of conceptual entities.
- A first step towards solving these tasks is the automated discovery of distributed symbol-like representations.
- Therefore we seek to split the input into separate entities and represent their information content efficiently, based on statistical regularities of the data that can be learned in an unsupervised fashion.
- Here we are concerned with the domain of images where entities naturally form groups of pixels (objects) that share mutual information.
- We are therefore interested in learning a perceptual grouping (or clustering) to recover these entities, and a corresponding structured representation that can later be used in a symbol-like fashion.

Effect of Hyperparameter K

Train			Test			Test Generalization		
# obj.	K	AMI	# obj.	K	AMI	# obj.	K	AMI
3	3	0.969 ± 0.006	3	3	0.970 ± 0.005	3	5	0.972 ± 0.007
3	5	0.997 ± 0.001	3	5	0.997 ± 0.002	3	3	0.914 ± 0.015
5	3	0.614 ± 0.003	5	3	0.614 ± 0.003	3	3	0.886 ± 0.010
5	5	0.878 ± 0.003	5	5	0.878 ± 0.003	3	5	0.981 ± 0.003

GitHub/SjoerdvanSteenkiste/Neural-EM

N-EM



A differentiable clustering procedure that learns a representation of a scene composed of primitive object represetations.

It consists of a spatial mixture model with K components that are parametrized by vectors $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K]$

A non-linear function f (a neural network) computes a distribution over images (factored across pixels) from $\boldsymbol{\theta}_k$.

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

For a fixed function f we can compute a Maximum Likelihood Estimate of $\boldsymbol{\theta}$ using generalized Expectation Maximization, which iteratively optimizes the expected data log-likelihood:

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

E-Step

Reassign the pixels to each cluster according to the posterior of $\mathbf{z}$

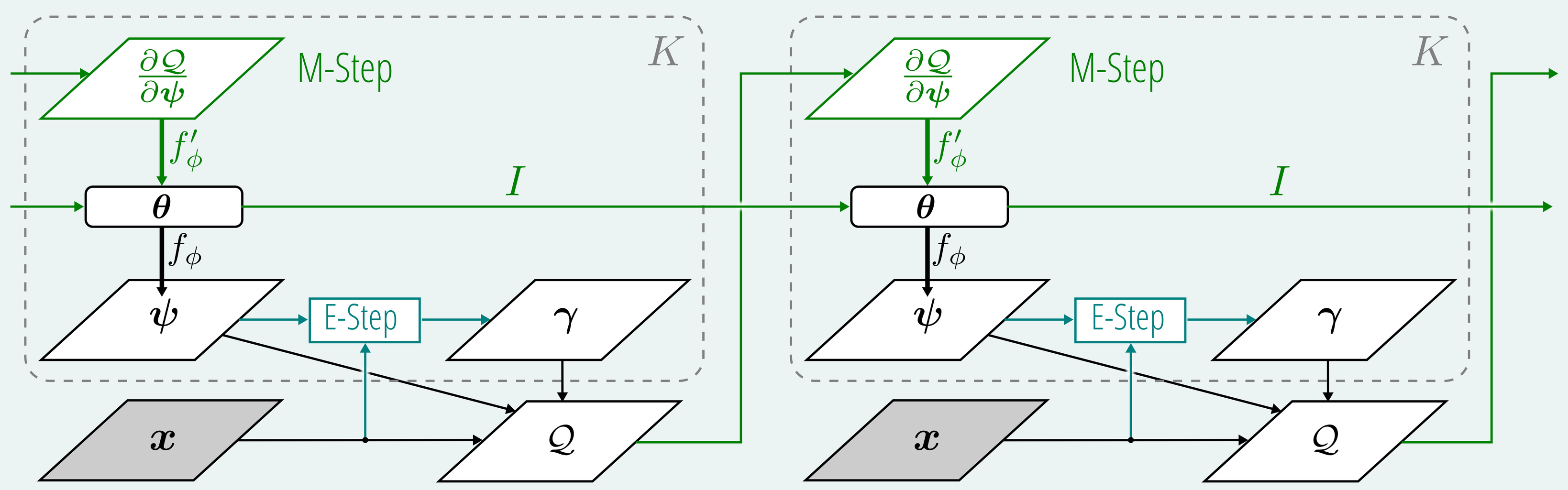
$$\gamma_{i,k} := P(z_{i,k} = 1|x_i, \psi_i^{\text{old}})$$

M-Step

Improve the expected value $\mathcal{Q}$ of the complete data likelihood by gradient ascent:

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

The unrolled gradient ascent updates form a computational graph that is end-to-end differentiable. We refer to this trainable procedure as Neural Expectation Maximization.



By relaxing the structure and converting the above graph into an RNN we obtain a more powerful version that we call **RNN-EM**

The statistical regularities required to cluster the pixels of an image into objects are encoded in the weights of the neural network, which we train to minimize a two part loss function:

$$L(\boldsymbol{x}) = -\sum_{i=1}^{D} \sum_{k=1}^{K} \underbrace{\gamma_{i,k} \log P(x_i, z_{i,k}|\psi_{i,k})}_{\text{intra-cluster loss}} - \underbrace{(1-\gamma_{i,k}) D_{KL}[P(x_i)||P(x_i|\psi_{i,k}, z_{i,k})]}_{\text{inter-cluster loss}}$$

The intra-cluster loss maximizes the data log likelihood (the same as for EM) and encourages each cluster to better reconstruct its pixels.

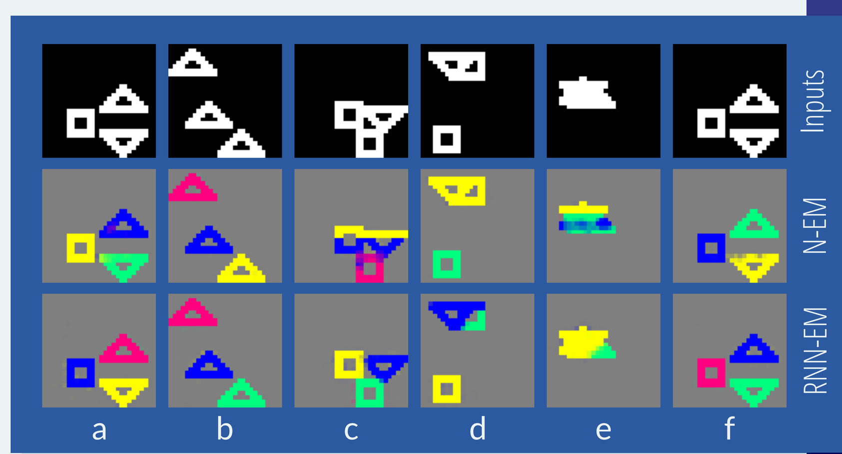
The inter-cluster loss minimizes the expected out-of-cluster data log likelihood. It encourages each cluster to specialize

Results

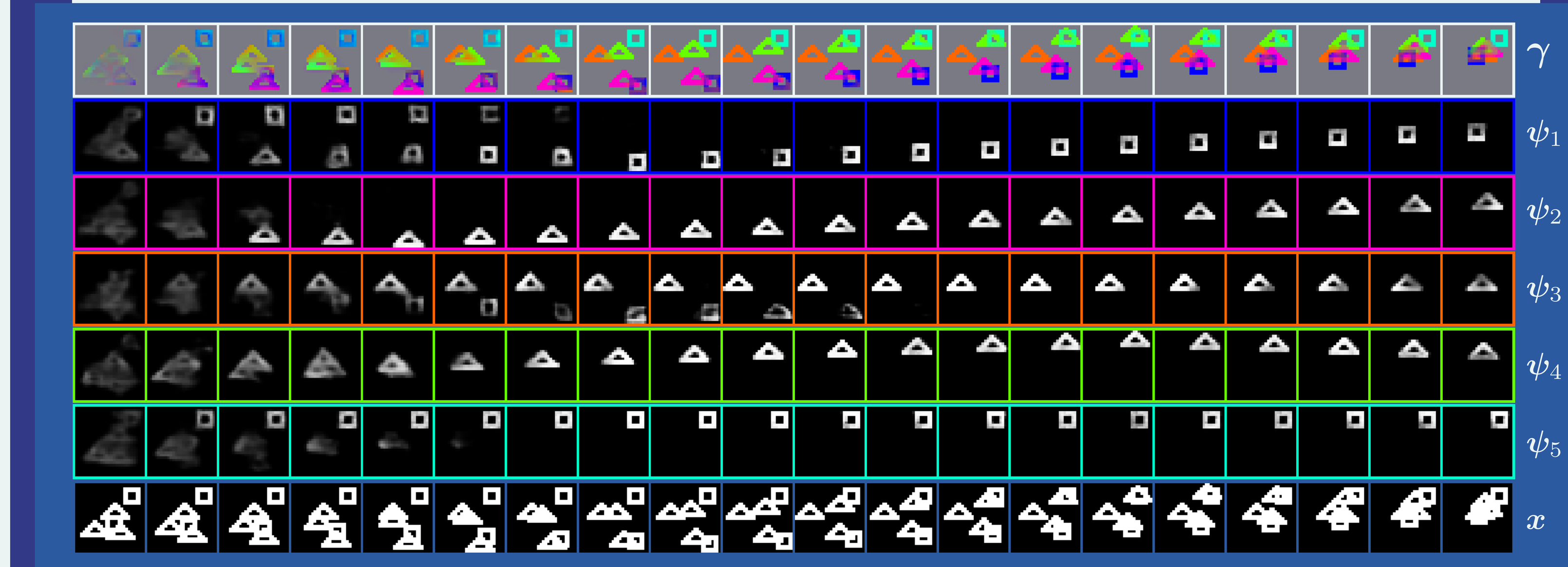
Shapes

RNN-EM and N-EM recover the invididual shapes accurately when they are separated (a, b, f), even when confronted with the same shape (b).

RNN-EM is able to handle most overlap (c, d) and only sometimes fails (e).

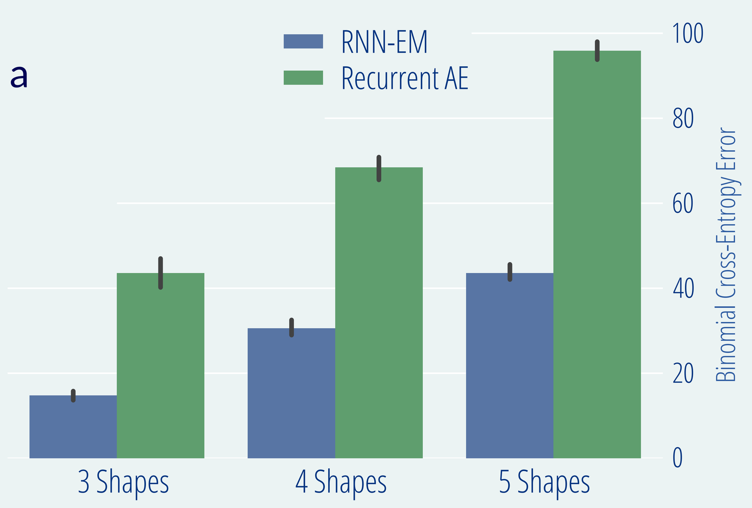


Flying Shapes (each shape moves in a random direction)



RNN-EM significantly outperforms a standard recurrent autoencoder in terms of next-step prediction on flying shapes with 3/4/5 shapes.

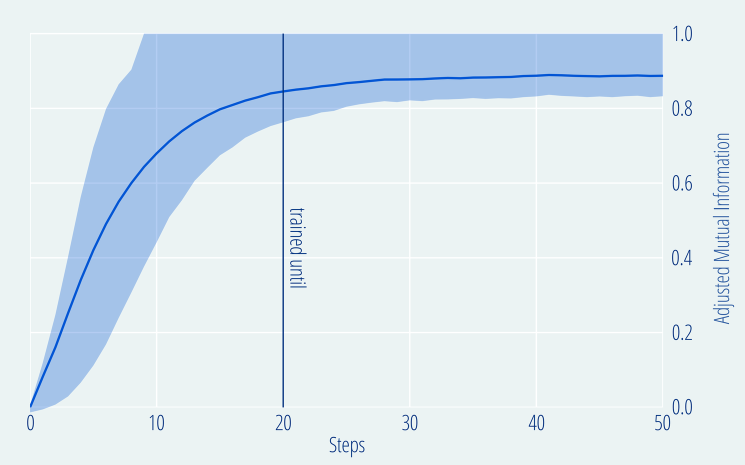
This highlights the fact that grouping is useful for next-step prediction.



Flying MNIST (each digit moves in a random direction)

Temporal coherence provides useful cues about the grouping of pixels.

The learned grouping dynamics are stable and generalize beyond the sequence-length on which they were trained.



* Authors contributed equally to this work.