- Multi-Object Representation Learning with Iterative Variational Inference
- Klaus Greff, Alexander Lerchner

Abstract

Previous work either focus on segmentation or representation learning. This work learns to segment and represent objects *jointly*.

Method

Multi-Ojbect Representation

Consider a dataset with each image composed of multiple objects,

- We should assume the existence of k latent variables $z = \{z_1, \ldots, z_k\}$.
- The likelihood is modelled as a mixture of gaussian, where **each component depends on exactly one latent variable**. That is

$$p(x|z) = \prod_{i=1}^{D} \sum_{k=1}^{K} m_{ik} \mathcal{N}(x_i | \mu_{ik}, \sigma_x^2)$$

Decoder Structure. The problem is to decode z_k to m_k and μ_k. We use **spatial broadcast network**. All slot share weights to ensure a common format.

Inference

The goal is to find good λ for $q_\lambda(z|x)$. The author proposed three difficulties

- Firstly, being a (spatial) mixture model, we need to infer both the components (i.e. object appearance) and the mixing (i.e. object segmentation).
- One slot may suffice. There is no reason that the inference procedure will model one object with one single slot. **Strong coupling may happend**.
- Slot permutation invariance induces a multimodel posterior with at least one mode per slot permutation. This means that **each permutation should be equally likely**. But VAE $q_\lambda(z|x)$ is uni-modal.

Iterative Inference. I'm still confused why this helps to tackle the above problem.

I think the most important idea is to update λ_k's **seperately**, using information specific to k. As in Iterative Amortized Inference, for each training example $x^{(i)}$ and k, they will start with a random guess λ_k, and iteratively optimize λ_k, using some information in the network. Since we are backpropagating into the parameter ϕ of the refinement network, we expect it learns how to optimize. This is how they do it:

$$z_k^{(t)} \sim q_\lambda(z_k^{(t)} | x)$$
$$\lambda_k^{(t+1)} \leftarrow \lambda_k^{(t+1)} + f_\phi(z_k^{(t)}, x, a_k)$$

Where a_k contains the following information:

- image $\boldsymbol{x}$, means $\boldsymbol{\mu}_k$, masks $\mathbf{m}_k$, and mask-logits $\hat{\boldsymbol{m}}_k$,
- gradients $\nabla_{\boldsymbol{\mu}_k}\mathcal{L}$, $\nabla_{\mathbf{m}_k}\mathcal{L}$, and $\nabla_{\boldsymbol{\lambda}_k}\mathcal{L}$,
- posterior mask $p(\mathbf{m}_k|\mathbf{x},\boldsymbol{\mu}) = \frac{p(x|\mu_k)}{\sum_j p(x|\mu_j)}$,
- pixelwise likelihood $p(\mathbf{x}|\mathbf{z})$,
- the leave-one-out likelihood $p(\mathbf{x}|\mathbf{z}_{i\neq k})$,
- and two coordinate channels like in the decoder.