Attend, Infer, Repeat: Fast Scene Understanding with Generative Models

S.M. Eslami,N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K.Kavukcuoglu, G. E. Hinton

Nicolas Brandt

nbrandt@cs.toronto.edu

Origins

Deep generative methods :

- \+ : Impressive samples and likelihood score
- \- : Lack of interpretable meaning

Structured generative methods :

- \+ : More easily interpretable
- \- : Inference hard and slow

How can we combine deep networks and structured probabilistic models in order to obtain interpretable data while being time efficient ?

Principle

Many real-world scenes can be decomposed into objects.

Thus, given an image **x**, we can make the modeling assumption that the underlying scene description **z** is structured into groups of variable $\mathbf{z}^i$.

Each $\mathbf{z}^i$ will represent the attributes of one object in the scene (type, appearance, position...)

Principle

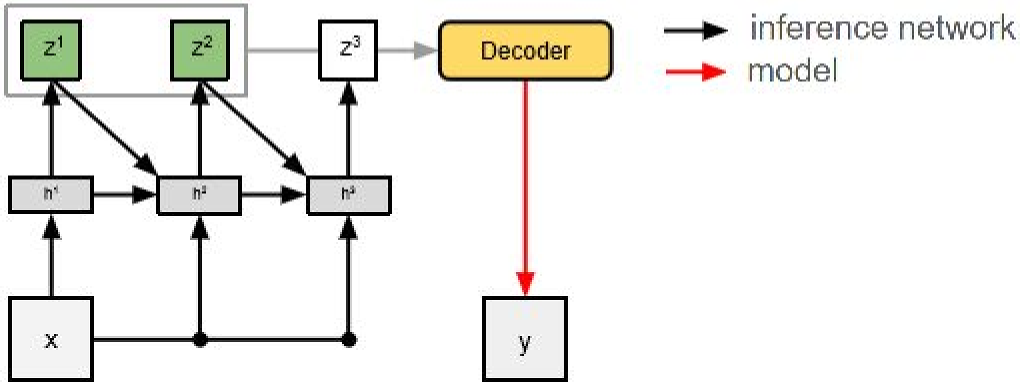
Given **x** and a model $p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$ parameterized by θ, we wish to recover **z** by computing $p_\theta(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})/p_\theta(\mathbf{x})$.

$p_\theta(\mathbf{x}) = \sum_{n=1}^{N} p_N(n) \int p_\theta(z|n)\, p_\theta(x|z)\, dz$ [1]

As the number of objects present in the image will most likely vary from a picture to another, $p_N(n)$ will be our prior on the number of objects.

NB: We have to define N which will be the maximum possible number of objects present in an image.

Principle



The inference network will attend to one object at a time and train it jointly with its model.

Inference Network

Most of the time, the equation [1] is intractable → Necessity to approximate the true posterior.

Learning a distribution $q_\Phi(\mathbf{z},n|\mathbf{x})$ parametrized by Φ that minimizes $KL[q_\Phi(\mathbf{z},n|\mathbf{x})||p_\theta(\mathbf{z},n|\mathbf{x})]$ (amortized variational approximation ~ VAE)

Nevertheless, in order to use this approximation we have to resolve 2 others problems.

Inference Network

Trans-dimensionality: Amortized variational approximation is normally used with a fixed size of the latent space, here it is a random variable.
→ We have to evaluate $p_N(n|\mathbf{x})=\int p_\theta(\mathbf{z},n|\mathbf{x})d\mathbf{z}$ for $n=1,...,N$

Symmetry: As the index for each object is arbitrary, we can see alternative assignments of objects appearing in an image **x** to latent variable $\mathbf{z}^i$.

In order to resolve these issues, we will use an iterative process implemented as a recurrent neural network. This network is run for N steps and will infer at each step the attributes of one object given the image and its previous knowledge of other objects on the image.

Inference Network

If we consider a vector $\mathbf{z}_{pres}$ composed of n ones followed by a zero we can consider $q_\Phi(\mathbf{z},\mathbf{z}_{pres}|\mathbf{x})$ instead of $q_\Phi(\mathbf{z},n|\mathbf{x})$.

This new representation will simplify the sequential reasoning : $\mathbf{z}_{pres}$ can be considered as a counter stop. While the neural network q_Φ outputs z_{pres}=1, it means that the networks should describe at least one more object, if z_{pres}=0, all objects have been described.

Learning process

The parameters θ (model) and Φ (inference network) can be jointly optimized by using gradient descent in order to maximize :

$$L(\theta,\phi)=E_{q_\phi}[\log\frac{p_\theta(\mathbf{x},\mathbf{z},n)}{q_\phi(\mathbf{z},n|\mathbf{x})}]$$ (negative free energy)

If p_θ is differentiable in θ, it is possible to compute a Monte Carlo Estimate of $\frac{\delta}{\delta\theta}L$.

Computing $\frac{\delta}{\delta\phi}L$ is a bit more complex.

Learning process

For a step i, we consider $w^i=(z^i_{pres},\mathbf{z}^i)$. Thus, by using chain rule, we have :

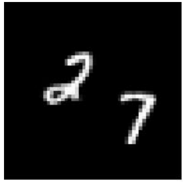
$$\frac{\delta}{\delta\phi}L=\sum_{i=1}^{N}\frac{\delta L}{\delta w_i}*\frac{\delta w_i}{\delta\phi} \quad .$$

Now, if we consider an arbitrary element z^i from $(z^i_{pres},\mathbf{z}^i)$, we will be able to compute the result with different methods depending on whether z^i is continuous (position) or discrete (z^i_{prez}).

Continuous: we use the ‘re-parametrization trick’ in order to ‘back-propagate’ through z^i

Discrete: we use the likelihood ratio estimator.

Experiment: MNIST digits



Objective: Learn to detect and generate the constituents digits from scratch.

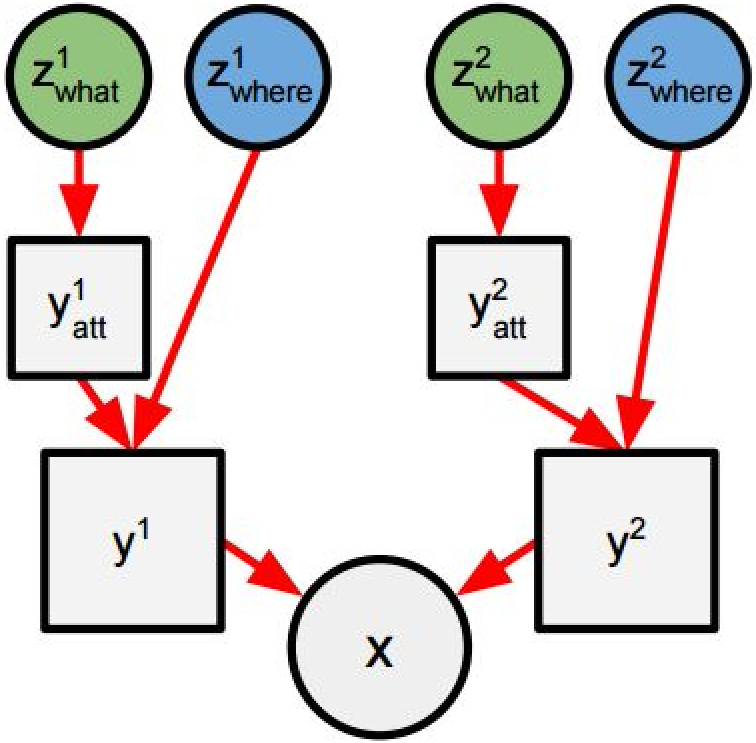
In this experiment, we will consider N=3.

In practice, each image will only contain 0,1 or 2 numbers.

Here, $\mathbf{z}^i = (z^i_{what}, z^i_{where})$ where z^i_{what} is an integer (value of the digit) and z^i_{where} is a 3-dimensional vector (scale and position of the digit)

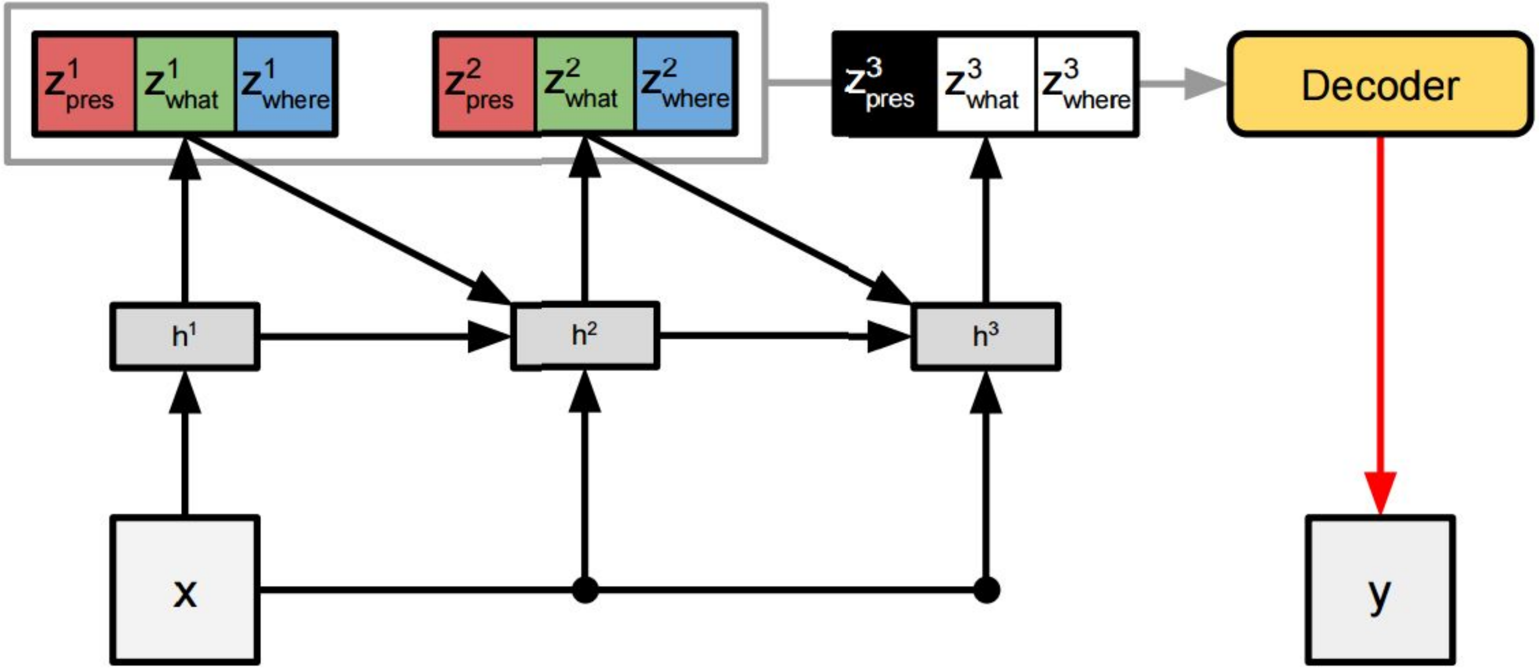
Experiment: MNIST digits

Generative Model:



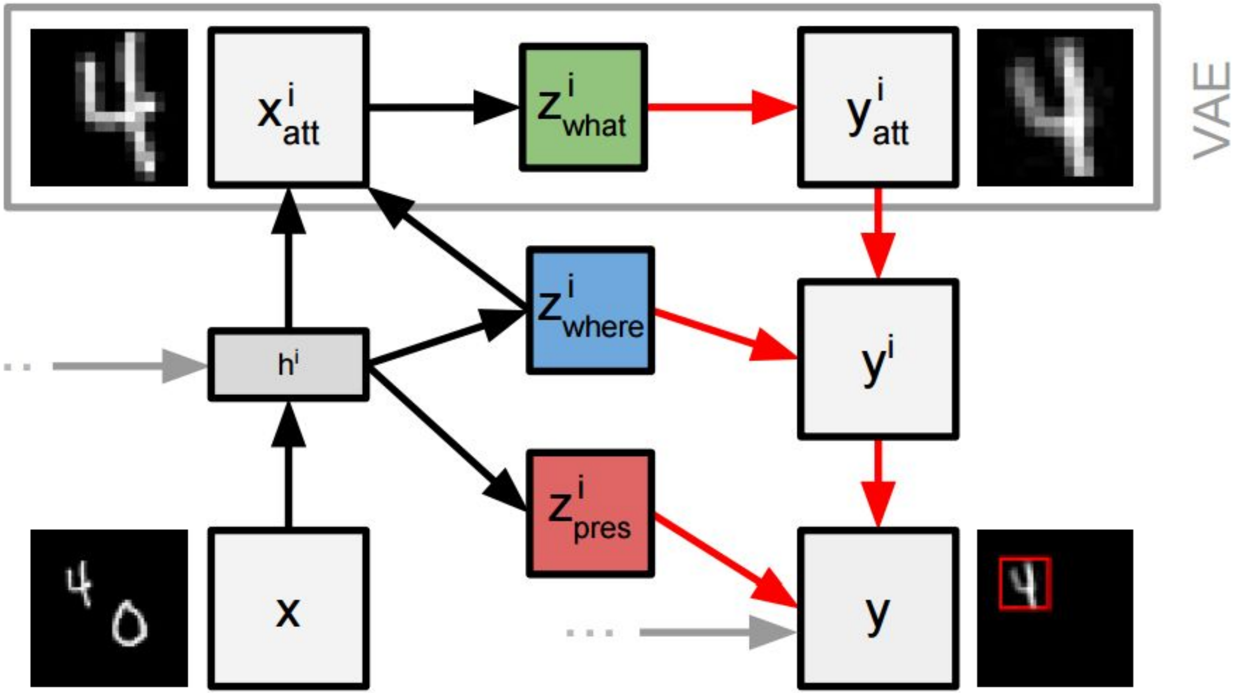
Experiment: MNIST digits

Inference Network:



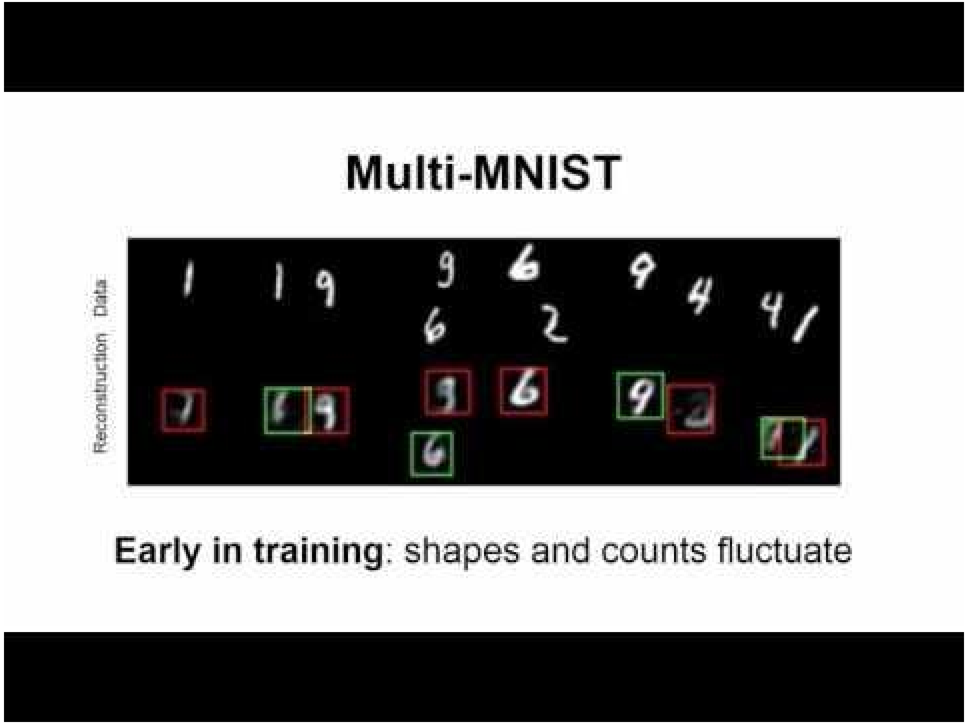
Experiment: MNIST digits

Interaction between Inference and Generation networks:



Experiment: MNIST digits

Result:



source : https://www.youtube.com/watch?v=4tc84kKdpY4&feature=youtu.be&t=60

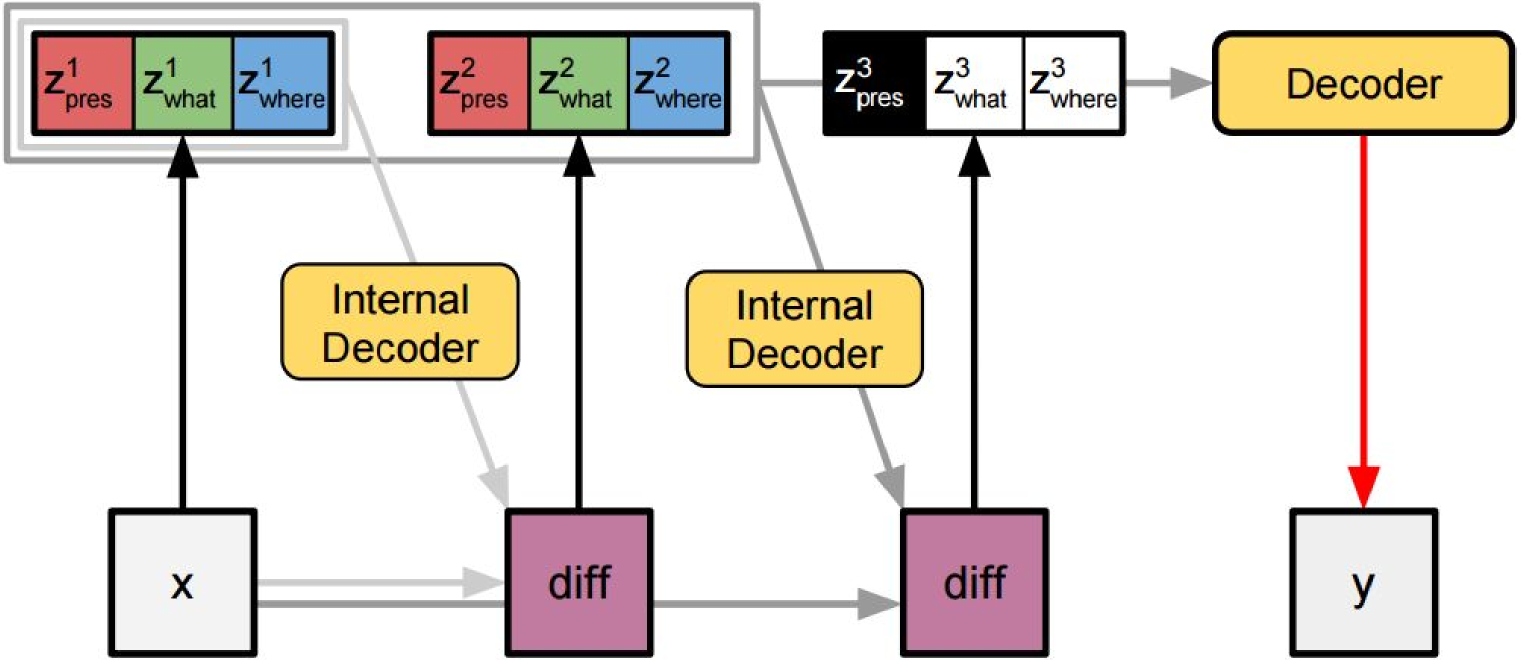
Generalization

When the model is trained only using images composed of 0, 1 or 2 digits, it will not be able to infer the correct count when given an image with 3 digits.

The model learnt during the training to not expect more than 2 digits

How can we improve the generalization ?

Differential AIR



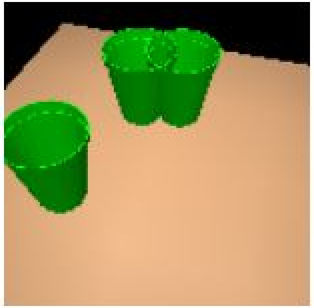
Conclusion

This model structure managed to keep interpretable representation while allowing fast inference (5.6 ms for MNIST).

Nevertheless, there are still some challenges :

- Dealing with the reconstruction loss

- Not limiting the maximum number of objects



Thank you for your attention !