ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Multi-model disentanglement of object representation

Master Thesis

Eric Stavarache

July 19, 2019

Advisors: Frederik Benzing, Asier Mujika

Department of Computer Science, ETH Zürich

Abstract

This example thesis briefly shows the main features of our thesis style, and how to use it for your purposes.

Contents

Chapter 1

Introduction

Humans reason about the surrounding world by decomposing it into orthogonal components. The idea of an object is decoupled from the qualities of the object: it is very easy for us to imagine a pink elephant, even though we have never observed such an animal, and these two words suffice for us to imagine a visual scene. The words are a latent representation of the scene.

Variational Autoencoders (VAE) [4] are a powerful framework for learning latent representations. Much work has already been done on disentangled representations, where we require that any two latent variables are uncorrelated.

Our work focuses on model-based disentanglement of objects. In particular, we would like to have one individual VAE which is trained to recognise and represent a single type of object: in a normal setting, we could have one VAE which represents chairs, another VAE which represents tables, and so on. We will call our approach the **SuperVAE**.

Most similar in spirit is the [2], where a single VAE model iteratively recognises objects from a scene by using attention masks. The main difference is that the MONet uses a single VAE for all objects, whereas we use multiple VAE's in parallel. Another similar idea is contained in [3], where they have a single VAE which tries to model the scene, and where they iteratively refine the latent parameters.

Chapter 2

Method

2.1 SuperVAE

The SuperVAE network tries to reconstructor a scene by feeding the image to some VAE's connected in parallel, whereby every VAE models a separate part of the scene.

Let us say that there are K VAE's, where each of them tries to model a different object. Then the image X will be fed independently to each of them.

Each VAE will then model a distribution over the latent variables, $q_{\theta_k}(\mathbf{z}_k|\mathbf{X})$.

By sampling from these distribution, they will generate two outputs of the same size as the image: $\hat{X}_k$, which is the k'th model reconstruction of the image, and $\hat{m}_k$, which is a confidence mask.

This confidence mask represents how sure a VAE is about its output for a given pixel. Higher confidence means that the pixel is part of an object which is of the type the VAE is modelling.

These confidence masks are produced by first having the models generate some "raw" confidence values, and then taking a pixel-wise softmax across all of the k models.

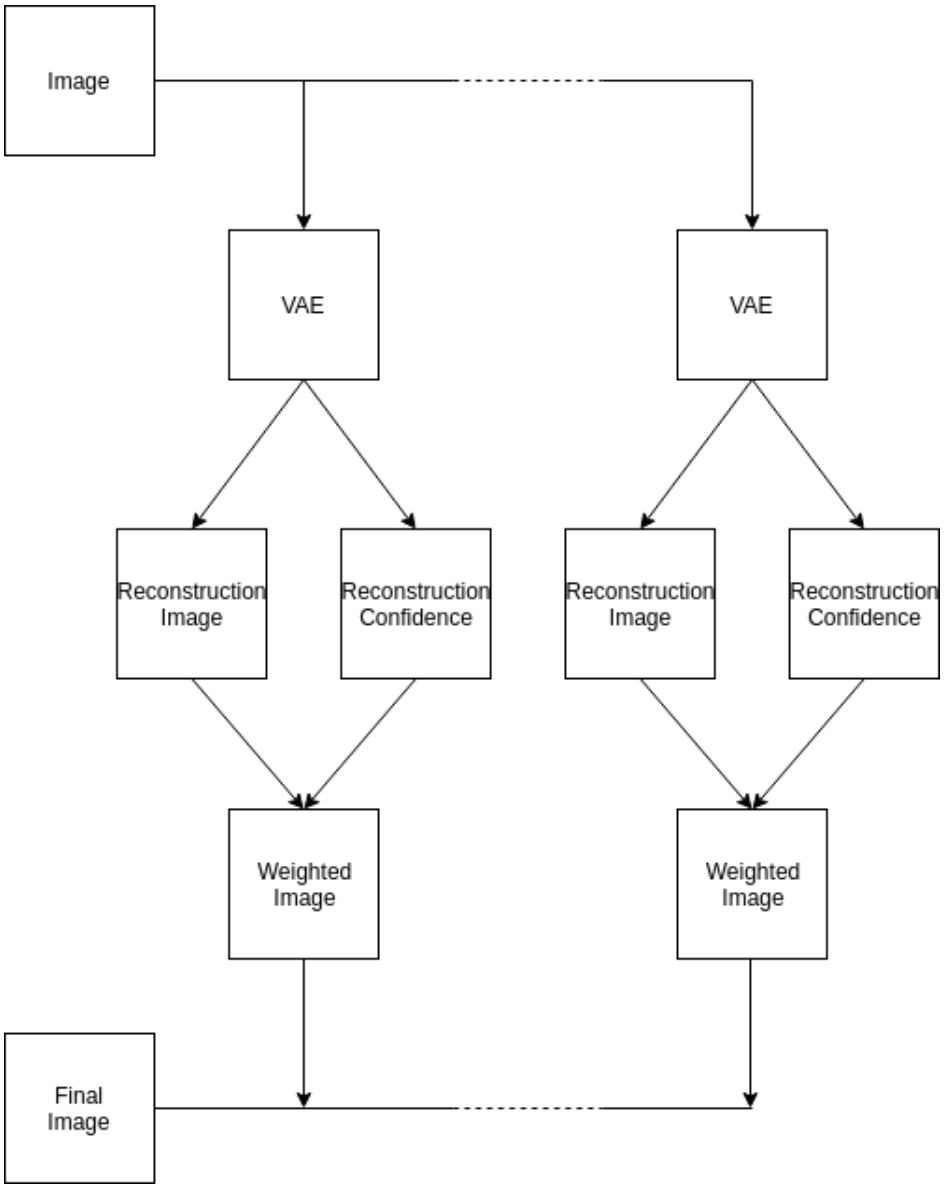
Thus, the masks are a probability distribution, where for each pixel we have what is the probability that it belongs to an object modelled by a VAE:

$\sum_{i=1}^{k} \hat{m}_k = \mathbf{1}$.

Our loss function is

$$\mathcal{L}(X;\theta_1,...,\theta_k) = \sum_{i=1}^{k} \|(\hat{X}_i - X) \odot \hat{m}_i\|_F^2 + \beta \sum_{i=1}^{k} D_{KL}[q_{\theta_k}(\mathbf{z}_k|\mathbf{X})||\mathcal{N}(0,\mathbf{I})] + \gamma \cdot -\log(\frac{1}{\log K} \cdot -\hat{m}\log\hat{m})$$

Figure 2.1: A diagram of how the model operates.



The first term of the loss is a L_2 loss weighted by each model's confidence. The intuition behind this is that the more responsability a VAE takes for drawing certain pixel, the more it should be penalized for making mistakes.

The second term is the standard KL loss term, weighted by a β hyperparameter, as first introduced in [1].

The last term is our original contribution, which is a cross-entropy loss. The idea is that the VAE's should try to be unassuming, and they should incur a cost if they decide to take on a lot of responsability for representing a pixel. We have the $-\hat{m} \log \hat{m}$ term, which is the normal cross-entropy, where we interpret the masks as component-wise distributions over VAE's. We divide it by the maximal value it can take ($\log K$, where K is the number of VAE's) in order to obtain a value between 0 and 1. We want to penalize situations where one VAE takes over everything, and in these situations the cross-entropy would be close to 0. As such, we take the negative log of the normalized cross-entropy. This whole term is weighted by another hyperparameter, γ.

We compare this with the approach of [2], where they have a single VAE which tries to model all objects, whereas we have a single VAE for each object type. Furthermore, in their model, the VAE is instructed what to model by the U-Net attention mask, whereas in our approach each VAE decides "independently" what to learn. In our approach, the only communication happening between the VAE's is via the softmax operation of the confidence masks.

Chapter 3

Results

For our experiments, we have taken a supervised approach to training the models.

3.1 MNIST

In order to test our idea, we began with the MNIST dataset. To make the task more challenging, and to allow decomposition via objects, we construct our training data by putting two digits side-by-side.

3.1.1 2-MNIST

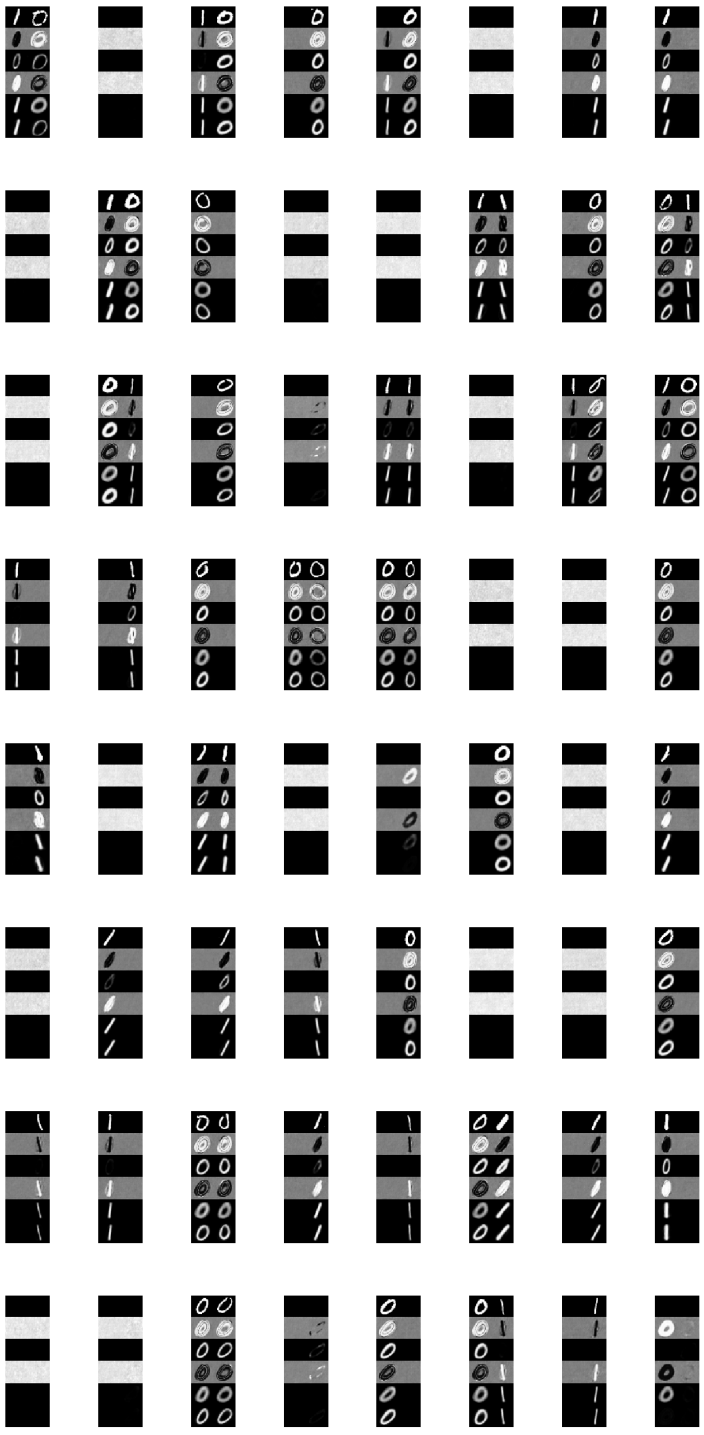
At the start, we set up only 2 VAE's in parallel, where we want that VAE-0 learns to recognise and model the digit 0, and VAE-1 models digit 1.

In order to accomplish this, we split the training into two stages:

1. In stage 0, VAE-0 trains on reconstructing images containing only the digit 0. The other model is active (so they are contributing to the confidence masks), but its weights are frozen, and only VAE-0 is learning.

2. In stage 1, VAE-1 trains on images containing digits 0 and 1 (so of the two slots, each digit is sampled independently). In this time, VAE-0 is frozen, *however* it is still contributing to the confidence masks. Because of this, places where the digit 0 appears are already assigned high confidence values by VAE-0, and so VAE-1 will not be motivated to learn them. On the other hand, places where digit 1 appears will not be recognised by VAE-0, and so VAE-1 will learn them.

Figure 3.1: Results of training on 2-MNIST. Each picture has 6 rows: first row is input image X, second row is confidence mask of VAE-0 $\hat{m}_0$, third row is reconstruction of VAE-0 $\hat{X}_0$, fourth row is $\hat{m}_1$, fifth row is $\hat{X}_1$, and last row is weighted reconstruction.



As we can see in figure 3.1.1, we have the desired effect: VAE 0 has high confidences (indicated by very white portions of the confidence mask) where the digit 0 appears, and has VAE 1 has high confidences where the digit 1 appears.

We note that there are cases when, even though VAE 0 has low confidence, it still draws a 0. This is because it was trained only on zeroes, so when encountering a digit 1 it does the only thing it knows how to do: output a 0. Similarly, VAE 1 draws very generic zeroes when the input digit is a 0. Since it was trained on both images of 0 and 1, it has had a chance to learn how a generic 0 looks like. Furthermore, it only takes one bit of information to store if the input is a 0 (which is then included in the KL-loss), but it has a lot to gain by making its reproduction closer to the original image (which is the first term in the overall loss function).

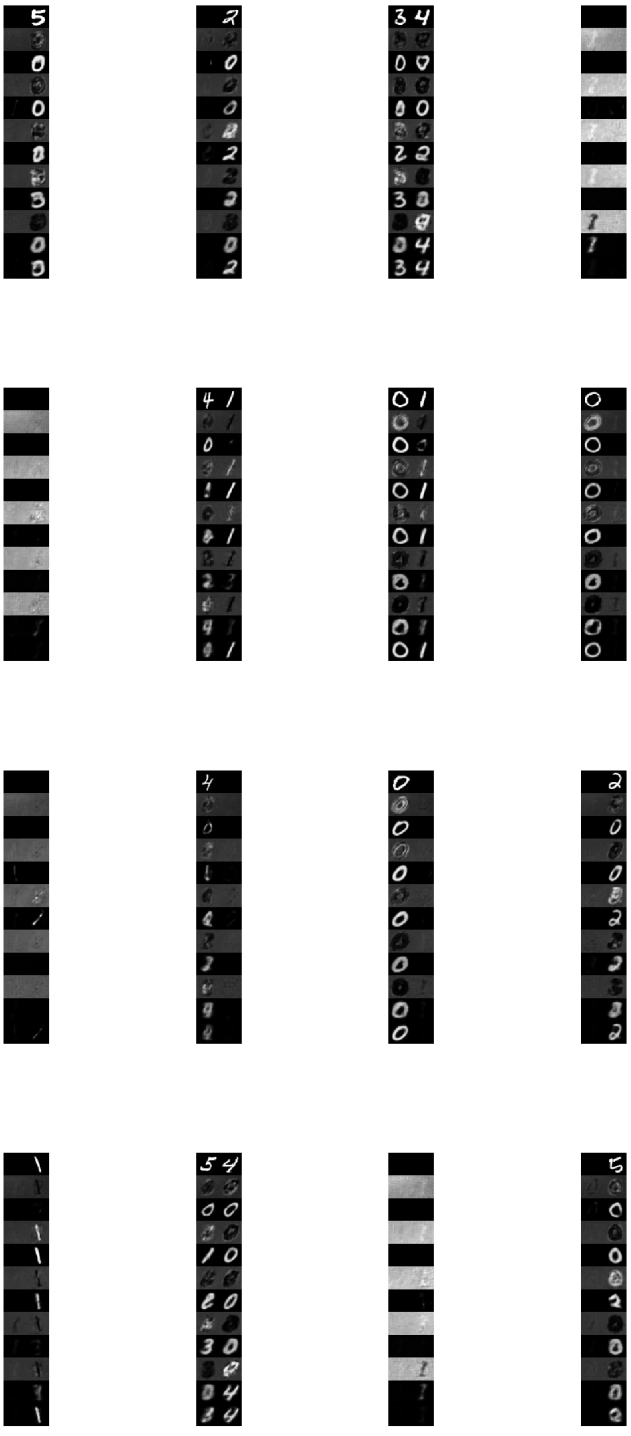
After this experiment, we then proceeded to see what would happen with 5 VAE's in parallel.

3.1.2 5-MNIST

For this dataset, we again have the same two digit side-by-side training data. However, we now have 5 VAE's wired up together, where again we would want that VAE i learns to model digit i.

The training is done analogous to 2-MNIST, in 5 stages, where in stage k model k is learning by looking at images containing digits $0, 1, ..., k$, and all of the other models are frozen (but still contributing to the confidences).

Figure 3.2: Results of training on 5-MNIST. Each picture has 12 rows: first row is input image X, last row is reconstructed image. In between, each adjacent pair of rows represents the confidences $\hat{m}_k$ and the reconstructions $\hat{X}_k$ for each of the $K = 5$ VAE's. Note that even though the digit 5 appears as part of the input, none of the models were trained on it: this is just test data. We include it only to see how models generalize to unseen digits.



From figure 3.1.2, we can see that the model achieves good reconstructions on the digits which it was trained on.

Again, we observe that when a digit 4 appears, only the last model has high confidences, and produces a good reconstruction; the other models generate a digit which they have learned, which minimizes the L_2 loss.

For digit 5, we observe that the model with the closest digit (in L_2 norm) has high confidence: sometimes the model for 2, and sometimes the model for 3.

Appendix A

Dummy Appendix

You can defer lengthy calculations that would otherwise only interrupt the flow of your thesis to an appendix.

Bibliography

[1] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv e-prints*, page arXiv:1804.03599, Apr 2018.

[2] Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *CoRR*, abs/1901.11390, 2019.

[3] Klaus Greff, Raphaël Lopez Kaufmann, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loïc Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. *CoRR*, abs/1903.00450, 2019.

[4] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114, Dec 2013.