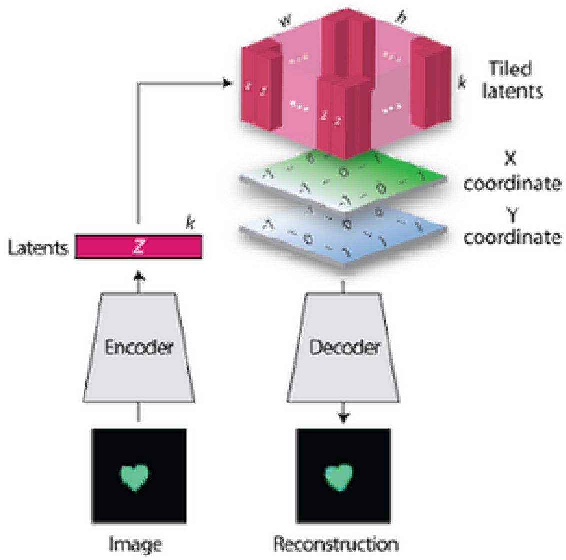
Disentanglement in Variational Autoencoder

A project documentation comparing different Variational Autoencoder



Lisa Golla
Luca Caspari
Melih Oezkan

UNIVERSITÄT OSNABRÜCK

A documentation of the final project for the course
Implementing Artificial Neural Networks With Tensorflow
WS 2021

Frequency band power & connectivity patterns - a basis for Alzheimer's disease diagnosis in EEG data?

—This paper presents the results of a reimplementation of scientific papers proposing methods to achieve a disentangled representation of data. Specifically, we implemented the Beta Variational Autoencoder, the Spatial Broadcast Decoder and the Hyperprior Variational Autoencoder. Focus of our result is especially laid on the disentanglement of the dataset images. The implementation could partially achieve the results presented by the papers. However, the models did not achieve an overall disentanglement, changes in several parameters of the latent dimension still influenced more than one generative factor in the reconstructed data. In order to make reproduction easily possible, you can find a link containing the code here.

Index Terms—**Deep learning, Variational Autoencoder, Spatial Broadcast Decoder, Beta Variational Autoencoder**

I. Introduction

In machine learning, it is a well-known challenge to improve generalization. In convolutional networks for instance, a good generalization is important for recognizing and distinguishing the component parts of an image. If a picture shows an object, the object is composed of different factors like position, color, shape, etc. Humans excel in recognizing the object and its generating components. The goal of Artificial Intelligence is to mimic the quick intuition process of a human in order to detect the components of objects in images and be able to create new objects using the generative factors.

The task of detecting the generative components of an image encompasses huge relevance in practice, since it can be used for different applications, for example in the medical field to detect anomalies or in the field of design to create new products. Although the task is still difficult for machine learning algorithms, there are already a lot of approaches to tackle the challenge. In this paper, we would like to approach this in the light of learning compositional, unsupervised representations. This seems rational, since a "compositional representation consists of components that can be recombined, and such recombination underlies generalization" [12]. Our paper focuses on the concept of compositional representation, also called disentanglement. This technique breaks down each feature into narrowly defined variables and afterwards, encodes them as separate dimensions. We studied the disentanglement of Autoencoders by employing a Beta and Hyper Variational Autoencoder and a Spatial Broadcast Decoder. We reimplemented these architectures while focusing on the ability to achieve a disentangled representation. For analyzing the structure of our network, we used the visualization tool TensorBoard.

II. Related Work

A. Variational Autoencoder

A standard Autoencoder maps input data onto a smaller vector, the latent dimension. This technique is for example useful to compress and reconstruct images. When it comes to the task of generating images, classical Autoencoders are not sufficient, because we do not have an influence on how the network represents the input data. The paper "Auto-encoding Variational Bayes" by D. P. Kingma and M. Welling proposed not to represent the input data as fixed values but as parameters for a statistical distribution. By forcing the representation to be similar to a Gaussian normal distribution, we can enforce that similar images are stored in similar vectors and thus gaining a meaningful representation of the dataset. This makes it possible to generate images. Since the backpropagation algorithm, used to train a neural network, runs into trouble when layers contain probabilistic processes, the latent dimension of a VAE does not actually represent a distribution but parameter vectors for the mean μ and standard deviation σ, which are then used sample values from a normal distribution [1].

B. Disentangled Variational Autoencoder

Even though the Variational Autoencoder learns a meaningful representation of the data, the latent vector is still entangled. A change of a single latent value affects multiple changes in the reconstructed image. Two papers, published by DeepMind in 2016 [2] and 2017 [3] gain a more meaningful representation by implementing the concept of disentanglement:

"[S]ingle latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors [...]." [2]

Inspired by neuroscientific research about the learning of ventral visual stream in infants, the authors aim to implement a deep learning model which is able to learn a representation of the input data, where each latent unit corresponds to a single generative factor of the input. In a dataset with faces for example, the change of one variable could have the effect that only the hair color changes. The authors propose a beta coefficient in the loss function in order to strengthen the similarity between the learnt distribution and the gaussian normal distribution. The papers show that the implementation of a beta constraint results in a more disentangled representation than the standard Variational Autoencoder [1] and outperforms other deep learning architectures in this regard. Important for learning disentangled factors is that the input was sampled from a continuous dataset [2]. Similar to infants learning to detect objects which change their position and orientation continuously, the factors of the data presented to the network need to change continuously as well. However, the papers also conclude that a stronger disentangled representation comes with a less qualitative reconstruction of the input factor, because the beta coefficient puts a stronger constraint on the representation. The model loses more information responsible for the image quality when passing the data through the network.

C. Spatial Broadcast Decoder

Due to the fact that disentanglement gains more and more importance, other approaches despite the Beta-VAE emerged to solve this problem.As one of these solutions, DeepMind suggests an architecture called Spatial Broadcast Decoder [12]. The architecture modifies the VAE decoder in order to find hyperparameters to robustly obtain disentangled representations from images. Moreover, the architecture allows to discriminate positional from non-positional features. A "normal" deconvolutional network has no spatial information of objects. Consequently, tasks like changing positions of objects are obviously difficult for the "normal" deconvolutional network, because the deconvolutional network has to learn complicated functions and has to propagate spatial symmetries through the network. The SBD solves this problem by removing all upsampling deconvolutions and instead broadcasts latent vectors across the space. Afterwards, it concatenates fixed X, Y coordinate channels and finally applies a fully convolutional network with 1x1 stride. Admittedly, the model also has to learn the encoded spatial information in its latent space to reconstruct the image, but it's actually using the encoded spatial information more efficiently. Indeed, there is one possible limitation. If the data does not take advantage of having access to an absolute coordinate system, performance could be hurt. As DeepMind exposed, their experiments ended up with the result that the SBD outperforms the VAE with the standard Decoder architecture. Additionally, the paper displays that the SBD can be combined with state-of-the-art models. In particular, the authors showed that a ß-VAE combined with a SBD performs better in terms of disentanglement and also learns a more efficient representation of the data than a ß-VAE without a SBD.

D. Hyperprior Induced Disentanglement

While the beta-VAE is able to create improved disentangled latent representations by constraining the distributions, this approach still leaves some issues. One of which is the impaired reconstruction of the original images. This impaired reconstruction occurs since the new introduced constraint reduces the importance of the reconstruction by giving more importance to the mutual information and KL-Divergence in general. To solve this problem and further improve the disentanglement of the beta-VAE the ELBO has been decomposed and modified in different ways [6]. Resulting in improved disentanglement and better reconstruction Another way of achieving slightly better results is by modifying the Gaussian distribution used for the prior and the latent representation. In their paper [4] introduce a new parameter Σ which decodes the covariance matrix for the Gaussian distribution. This parameter is sampled from a Wishart distribution. However, any other fitting distribution can be used instead. Which is then used as a parameter for a multivariate normal distribution from which the latent vector z is sampled. Creating an additional constraint and thus making the network more likely to incorporate disentanglement whilst having a better reconstruction. By using the generated covariance matrix as an input for the multivariate normal Gaussian we allow the Gaussian to select its values more freely. This enforces a greater disentanglement in general while also allowing for a dimension wise variance giving the latent dimension the ability to capture correlated properties. And since we don't have the beta constraint which increases reconstruction error the resulting images should be better whilst being more disentangled.

III. Implementation

A. Datasets

For our implementations, we created an image generator similar to the "synthetic binary dataset of [...] 2D shapes" introduced in the papers of the Beta-VAE [2] [3]. Our images contain ellipses generated by the four generative factors: position X (16 values), position Y (16 values), scale (6 values) and orientation (40 values). The dataset is ordered in such a way that the generative factors of the ellipses transform continuously. Furthermore, we used the Fashion-MNIST Dataset, in order to check qualitatively the reconstruction of the network.

IV. Models and Results

A. Beta Variational Autoencoder

Our model of the Beta Variational Autoencoder is a modification of the Variational Autoencoder Framework implemented on the Tensorflow Tutorial Webpage. The Encoder of our network architecture consists of a Convolutional Layer with 32 kernels, followed by two Convolutional Layers with 64 kernels. The input is then flattened and pushed through two separate Dense Layers producing the parameter vectors for the latent distribution. The reparameterization function of the main class "BetaVAE" coordinates the sampling process of the latent distribution. The Decoder receives the input by a Dense layer and reshapes it to images. It is then composed of two Transposed Convolutions with 64 kernels, followed by a Transposed Convolution with 32 kernels. By pushing the input through a Transposed Convolutional Layer with one filter, the network outputs one image.

We also tried architectures with a larger number of kernels, but none of them performed better. Adding a Dropout Layer and Batch Normalization led to slightly worse results, which is why we abstained from using optimization techniques. Since an analysis of the weights ensured that many parameters of the network do not have values clustered around zero, we added a slight L2 regularization to most of the layers.

The loss function represents constraint optimization problem [2] [3]. We minimize the loss of the reconstruction as well as the KL-Divergence measuring the similarity of the latent distribution and the normal distribution:

$$\mathcal{L}(\theta, \phi, x) = \mathbb{E}_{q\phi(z|x)}[log_{p\theta}(x|z)] - \beta D_{KL}(q_{\phi}(z|x) \,||\, p(x))$$

The Beta Variational Autoencoder focuses on ensuring a good representation by adding a beta constraint of 4. Tuning the beta coefficient to 1 returns the model into the classical variational autoencoder. So, for examining the impact of the beta constraint on the training results, we used the Beta Variational Autoencoder with beta = 1. We trained the Beta Variational Autoencoder for 50 epochs with a batch size of 32 and a beta coefficient of 4. Looking at the loss and the reconstructed image, we can see that the network performed well in reconstructing the image:

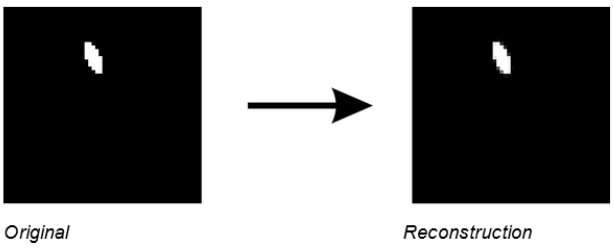


Fig. 1. A picture of the orginal and the reconstructed picture

The figure IV-A plots the latent representation the network has learnt. Every line in the plot displays the change of one latent parameter while keeping the rest of the latent dimension constant.

Concerning the paper, we should see a more disentangled representation the higher the beta coefficient is. So the change of a single latent parameter should change one of the generative factors position X, position Y, scale and rotation. The following table contains the latent dimension with beta=4, where we interpreted that it might represent a generative factor:

generative factor	**latent dim** (numerated from top to bottom)
position X	z7, z10,
position Y	z3, z6, z9
scale	z1, z3, z6, (z4, z5, z8)
rotation	z3, z4, z10

Fig. 2. The generative factors assigned to the corresponding latent dimensions

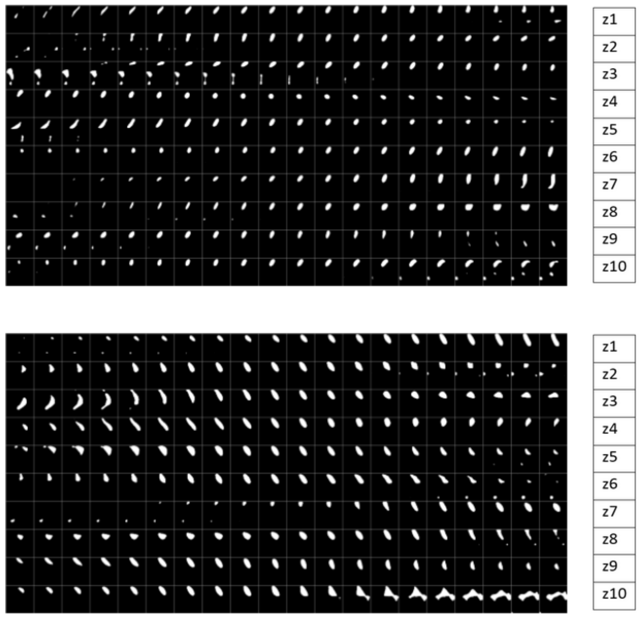


Fig. 3. Latent representation of Top : VAE (b=1) and Bottom : Beta-VAE (b=4)

We can deduce that the Beta-VAE learnt a more disentangled representation of the data than the standard Variational Autoencoder. The generative factors are easier to recognize, f.e. the rotation in z4. Unfortunately, our model does not seem to learn an overall good disentangled representation, since f.e. z3 represents three generative factors. Furthermore, the scale is dependent on a majority of the latent parameters, so it is difficult to change one of the generative factors without changing the scale. We trained the network with beta=6, too, but the network did not produce a more disentangled representation.

B. Spatial Broadcast Decoder

When reimplementing the SBD we used the resources [9] [10] as an orientation and compared the implementations with the description in the paper [12]. Since a Spatial Broadcast Decoder is a VAE with a changed Decoder architecture, we decided to reuse the VAE structure already used for the Beta VAE and changed the de - and encoder. The Encoder of our network architecture consists of 4 Convolutional Layers with 64 kernels The input is then flattened and pushed through two separate Dense Layers producing the parameter vectors for the latent distribution.

The Spatial Broadcast Decoder replaces the upsampling convolutions of the deconvolutional architecture by tiling the latent code z across the original image space. The function tf.tile is used here. By employing tf.linspace, fixed X,Y channels are defined and afterwards changed to a list of 2-D coordinate arrays for evaluating expressions on a 2-D grid with tf.meshgrid. Finally, tiled latents and fixed coordinate channels are concatenated and applied to unstrided convolutional layers.

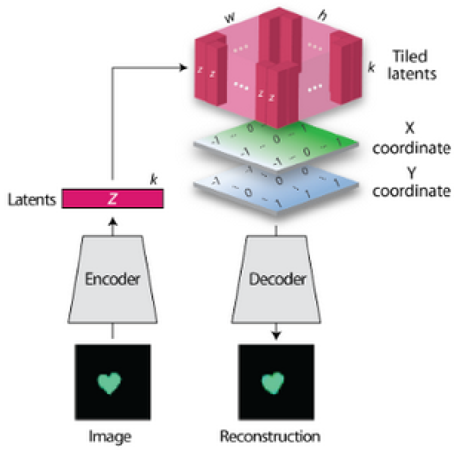


Fig. 4. Encoder Decoder Model

For a deeper understanding, the graphic on the right is quite expressive showing these coherences graphically [12].

Concerning the unstrided convolutional layers we used 3 convolutional layers each with kernel size (3,3) and a filter size of 64 for the first 2 layers and a filter size of 1 for the last layer. We decided on this architecture, since it performed best on our given dataset compared to other architectures.

One reason to avoid upsampling convolutions is that they are susceptible to checkerboard artifacts leading to a constrained reconstruction accuracy [12]. As further hypothesized, this may hurt disentanglement performance in the latent space. Another reason is that "normal" (de-)convolutional layers do not perform well at learning coordinate transformation, because the filters do not learn information about their position. In order to overcome complicated functions for coordinate transformations, the approach of appending coordinate channels is used. Actually, it turned out that this approach increases performance in several cases, since often coordinate transformation is implicitly needed [13].

We trained the Spatial Broadcast Decoder for 20 epochs with a latent dim of 10. The following image plots the reconstruction dependent on the latent dimension.

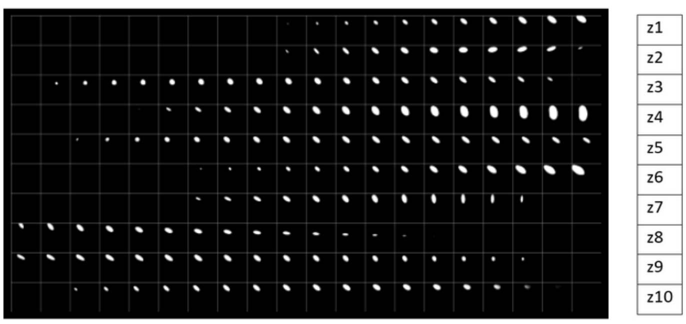


Fig. 5. Visualized reconstruction of the Spatial Broadcast Decoder

We can summarize that our network did not reach results of the paper with the same strength. For several latent parameters, there was no direct, obvious, learning achievement for a single generative factor. However, some latent dimensions learnt specific features of the image. z7 seems to represent the orientation of the ellipse and a change of z4 induces a strong change in the scale of the object. z8 seems to vary the y-position of the ellipse.

Generative factor	latent dim (numerated from top to bottom)
position X	z10
position Y	z8
scale	z1, z4, z6, z7, z9 (z3, z2)
rotation	z2, z3, z5, z6 (z7, z4, z8, z10)

Fig. 6. Generative factors assigned to respective latent dimension for the Spatial Broadcast Decoder

Lastly, we present our results for the Spatial Broadcast Decoder combined with the Beta variational Autoencoder loss function. Again we trained it for 20 epochs with 10 latent dimensions.

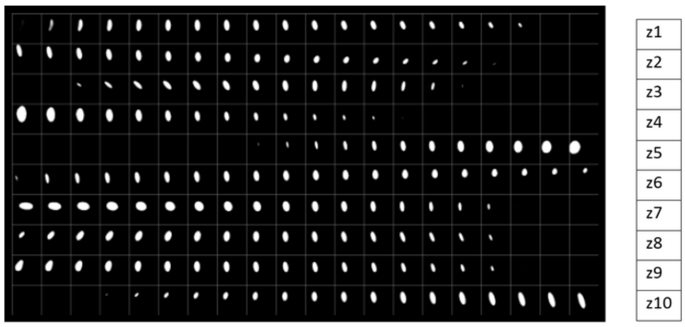


Fig. 7. Reconstruction of the Spatial Broadcast Decoder combined with the Beta variational Autoencoder loss function

The paper [12] proposed that the combination of both would lead to a better performance and disentanglement. Indeed, one can observe, when directly contrasting both performances that for the combination implementation, there are clearly more latent dimensions, which are only dependent on one generative factor. We would like to emphasize the relative good quality of the disentanglement compared to our implementation only using the SBD. Namely, the latent dimensions z4, z5, z7, z8, z9 are more meaningful and expressive for their learning achievements. Latent dimension z4 for example seems to represent a strong change in the scale, dimension 6 seems to vary the y-position of the ellipse and 8 seems to represent the orientation of the ellipse.

Nevertheless, our results do not measure up to the given results of the paper, especially the generative factor for the x-position of the object was hardly learned by our model.

Generative factor	latent dim (numerated from top to bottom)
position X	
position Y	6, (2)
scale	1, 2, 4, 5, 7, 9, 10 (6)
rotation	3, 8 (1, 10)

Fig. 8. Generative factors assigned to respective latent dimension for the Spatial Broadcast Decoder combined with the Beta variational Autoencoder loss function

C. Hyperprior VAE

For implementing the hyperprior VAE we used the paper [4] and sources from the official GitHub repository of the authors [14]. For the network architecture we used a simple Encoder and Decoder structure with the given values from the paper. This network however has a slightly different Encoder. The Encoder does not only output a sigma but modifies it through some calculations, so it takes the shape of a sample from a Wishart distribution.

Since we don't sample from a single normal distribution anymore we introduce a multivariate Gaussian distribution for which our encoder network outputs the needed parameter mu and its covariance matrix. We then sample from this multivariate Gaussian and run it through the Decoder network. From which we then get our recreated image.

Additionally, since we use a Wishart distribution to calculate the covariance matrix Σ and thus change the prior p(Σ) and calculate our sample using this distribution the formula for the latent representation p(x—z) changes respectively. Which is why the loss function has to be changed too.

V. Conclusion

Our main aim was to train different Variational Autoencoder architectures on a continuous dataset, which was generated by distinct factors. We expected to receive a disentangled latent representation. In the Spatial Broadcast Decoder and the Beta Variational Autoencoder, some latent parameters indeed seem to learn single generative factors of the data. We interpret this as a proof for the functionality of the approach. In our implementations, the Beta-VAE outperforms a VAE and the combination of an SBD and a Beta-VAE outperforms a normal SBD. This aligns with the hypotheses of the papers by DeepMind. Unfortunately, the disentanglement in the implemented architectures did not succeed entirely. The change of some single latent parameter usually affects more than one change in the reconstruction image. We hypothesize that one reason for this performance could be that the Ellipse Dataset does not perform the correct continuous transformations in order to let the models detect the generative factors. For the network, the transformations probably seem dependent on each other, since all factors change at the same time. Although the paper by Abdul F. A., Harold Soh [4] suggested that a Hyperprior VAE further improves disentanglement and reconstruction; we were not able to achieve such a result either by using unsuccessful seeds or some other unknown hyperparameters. This leaves the question open if improved disentanglement can be achieved by adding a prior to the network.

VI. Pictures title page

University Osnabrück Logo

References

[1] D. P. Kingma, M. Welling. Auto-encoding variational bayes. arXiv:1312.6114, 2014.

[2] I. Higgins, L. Matthey, X. Glorot, A. Pal, B. Uria, C. Blundell, S. Mohamed, A. Lerchner: Early Visual Concept Learning with Unsupervised Deep Learning. arXiv:1606.05579, 2016

[3] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner: beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. OpenReview, 2017

[4] Abdul F. A., Harold Soh: Hyperprior Induced Unsupervised Disentanglement of Latent Representations, arXiv:1809.04497v3, 2019

[5] Emile M., Tom R., N. Siddartha, Yee W. T.: Disentangling Disentanglement in Variational Autoencoders. arXiv:1812.02833 2019

[6] Ricky T. Q. Chen and Xuechen Li and Roger Grosse and David Duvenaud: Isolating Sources of Disentanglement in Variational Autoencoders, arXiv:1802.04942 2019

[7] Hiroshi T., Tomoharu I., Yuki Y. Masanori Y., Satoshi Y.: Student-t Variational Autoencoder for Robust Density Estimation, Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018

[8] Alexej K., (02 December 2019) Learning Hierarchical Priors in VAEs. Retrieved from https://argmax.ai/blog/vhp-vae/

[9] L.Binden. , (09 March 2019) Spatial Broadcast Decoder Implementation https://github.com/lukaszbinden/spatial-broadcast-decoder/blob/master/sbd.py 4.4 2021

[10] M. Norden , (25 September 2020) Spatial Broadcast Decoder Paper summary `https://borea17.github.io/paper_summaries/spatial_broadcast_decoder#visualization-functions` 4.4.2021

[11] T. Bepler., E. D. Zhong, K.Kelley, E. Brignole, B. Berger ,(25 September 2019)Explicitly disentangling image content from translation and rotation with spatial-VAE *1909.11663v1.pdf (arxiv.org) 4.4.2021

[12] N. Wattens, L. Matthey., C. P. Burgess, A. Lerchner. ,(14 August 2019) Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs *1901.07017v2.pdf (arxiv.org) 4.4.2021

[13] R. Liu., J. Lehman. , P. Molino., F. P.Such, E. Frank., A. Sergeev, J. Yosinski (03 December 2018) An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution https://arxiv.org/abs/1807.03247 4.4.2021

[14] Ansari, Abdul Fatir and Soh, Harold, CHyVAE, GitHub repository, https://github.com/clear-nus/CHyVAE