EXAM PROJECT FOR PML 2022/2023

REPORT

Christian Dybdahl Troelsen
Department of Computer Science
University of Copenhagen
Universitetsparken 1
DK-2100 Copenhagen Ø
tfp233@alumni.ku.dk

Jens Sørensen
Department of Computer Science
University of Copenhagen
Universitetsparken 1
DK-2100 Copenhagen Ø
qrw992@alumni.ku.dk

Mathias Rasmussen
Department of Computer Science
University of Copenhagen
Universitetsparken 1
DK-2100 Copenhagen Ø
tjc725@alumni.ku.dk

January 18, 2023

1 Density modeling

1.1 Implement a convolutional VAE

Architecture of the convolutional VAE Our implementation of the convolutional VAE is similar in architecture to the original VAE. The primary difference between the two models is that the two linear layers in the encoder and decoder part of the original VAE have been replaced by respectively two convolutional layers and two transposed convolutional layers in the convolutional VAE. The two convolutional layers in the encoder part of the convolutional VAE both use kernels of size 3×3, 1×1 padding and a stride of 2×2. The first convolutional layer has 1 input channel and 16 output channels, while the second convolutional layers has 16 input channels and 32 output channels. The transposed convolutional layers in the decoder part of the convolutional VAE complement the aforementioned convolutional layers. Both transposed convolutional layers use kernels of size 3×3, 1×1 padding 1×1, output padding and a stride of 2×2. The first transposed convolutional layer has 32 input channels and 16 output channels, while the second transposed convolutional layers has 16 input channels and 1 output channel. Conceptually, the encoder part of the convolutional VAE condenses its input images $\boldsymbol{x}$ of size $D = N \times N$ into a compact but feature rich representation $\boldsymbol{z}$ of size $C = 2$, while the corresponding decoder "unfolds" this compact representation in reverse order, ultimately producing output images $\boldsymbol{y}$ of the same size as $\boldsymbol{x}$.

Parameter estimation In order to optimize the convolutional VAE model we estimate the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ that maximize the ELBO, which is a lower bound on the log likelihood of the data $\boldsymbol{x}$ and is defined as $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z}))$. This is the same criterion used to optimize the original VAE. In particular, both models use the reparameterization trick to sample latent variables $\boldsymbol{z}$ from a factorized gaussian approximate posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{u}, \mathrm{diag}(\boldsymbol{\sigma}^2))$, so the Kullback-Leibler divergence, which acts as a regularization term in the ELBO, simplifies to $\frac{1}{2}\sum_{d=1}^{D}(\sigma_d^2 + \mu_d^2 - 1 - \ln(\sigma_d^2))$, assuming that the prior over $\boldsymbol{z}$ is also a diagonal gaussian $p_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$. Here $\ln \boldsymbol{\sigma}^2$ and $\boldsymbol{\mu}$ are the outputs of the encoder when applied to $\boldsymbol{x}$. In order to compute the first term in the ELBO, which is the expected likelihood of $\boldsymbol{x}$ given $\boldsymbol{z}$, we utilize the output $\boldsymbol{y}$ of the decoder. We fed $\boldsymbol{y}$ fed through a sigmoid activation layer, which produces the mean parameters $\boldsymbol{p}$ of a factorized multivariate Bernoulli distribution $\prod_j^D \mathrm{Bernoulli}(x_j; p_j)$ acting as the likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ of the input $\boldsymbol{x}$ given latent variable $\boldsymbol{z}$. We then approximate the expected log likelihood by computing the negative of the binary cross entropy of $\boldsymbol{p}$ and $\boldsymbol{x}$, where $\boldsymbol{x}$ is treated as a set of probability values. Given the simplified ELBO, we optimize the parameters of both models using a variant of gradient descent, namely the Adam algorithm with a learning rate of 0.001 and a batch size of 128. In order to ensure convergence we run the Adam algorithm for a total of 25 epochs for each model. We train both models on the complete MNIST training set.

Performance of VAE models We compare the performance of the two VAE models on the MNIST test set using two quantitative measures and three qualitative experiments. The two quantitative measures are the mean log likelihood and the MSE loss. The three qualitative experiments are

1. Clustering MNIST test data in latent space by mapping each observation $\boldsymbol{x}_i$ to corresponding mean parameters $\boldsymbol{\mu}_i$ using the encoder and subsequently plotting these $\boldsymbol{\mu}_i$ with colors based on the actual labels t_i associated with each $\boldsymbol{x}_i$.

2. Exploring the latent space by sampling latent variables $\boldsymbol{z}$ deterministically from a regular grid, then using the decoder to map these $\boldsymbol{z}$ to corresponding mean parameters $\boldsymbol{p}$, and finally visualizing these $\boldsymbol{p}$ as images.

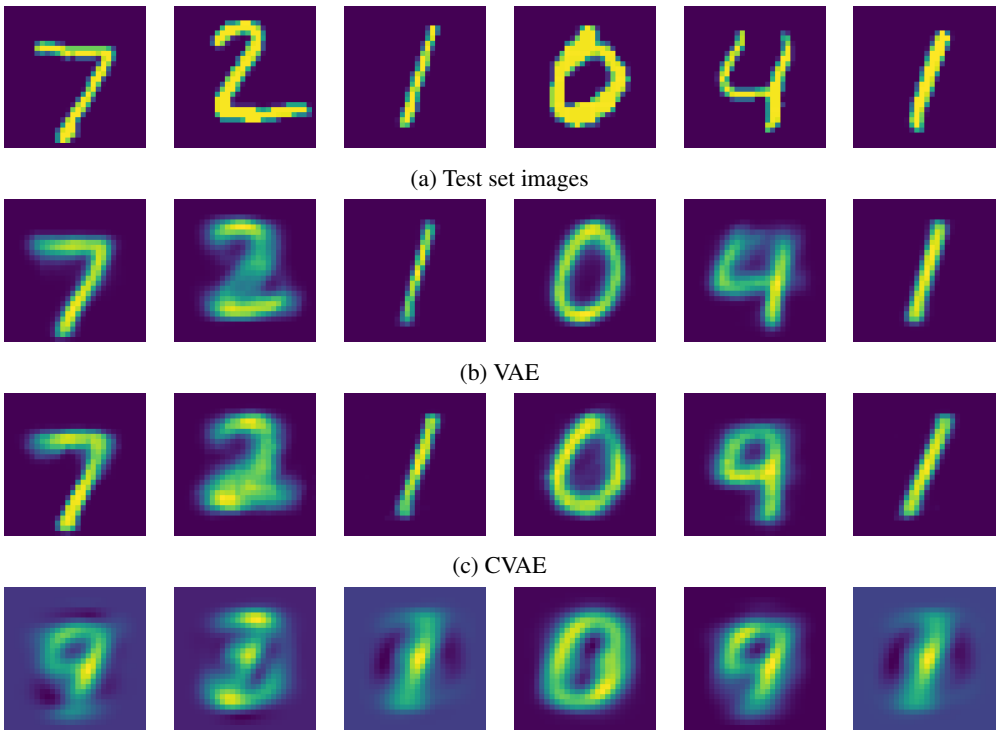
3. Reconstructing selected observations $\boldsymbol{x}_i$ from the MNIST test set by mapping these to posterior distributions $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$, then randomly sampling $\boldsymbol{z}_i$ from these posterior distributions, and finally using the decoder to map these $\boldsymbol{z}_i$ to corresponding mean parameters $\boldsymbol{p}_i$, which are finally interpreted as reconstruction of the inputs $\boldsymbol{x}_i$.

A comparison of the mean log likelihood and MSE loss for the two VAE models is shown in Table 1. As can be seen,

1.2 Alternative models

1.2.1 Probabilistic PCA

1.2.2 Gaussian Mixture Model



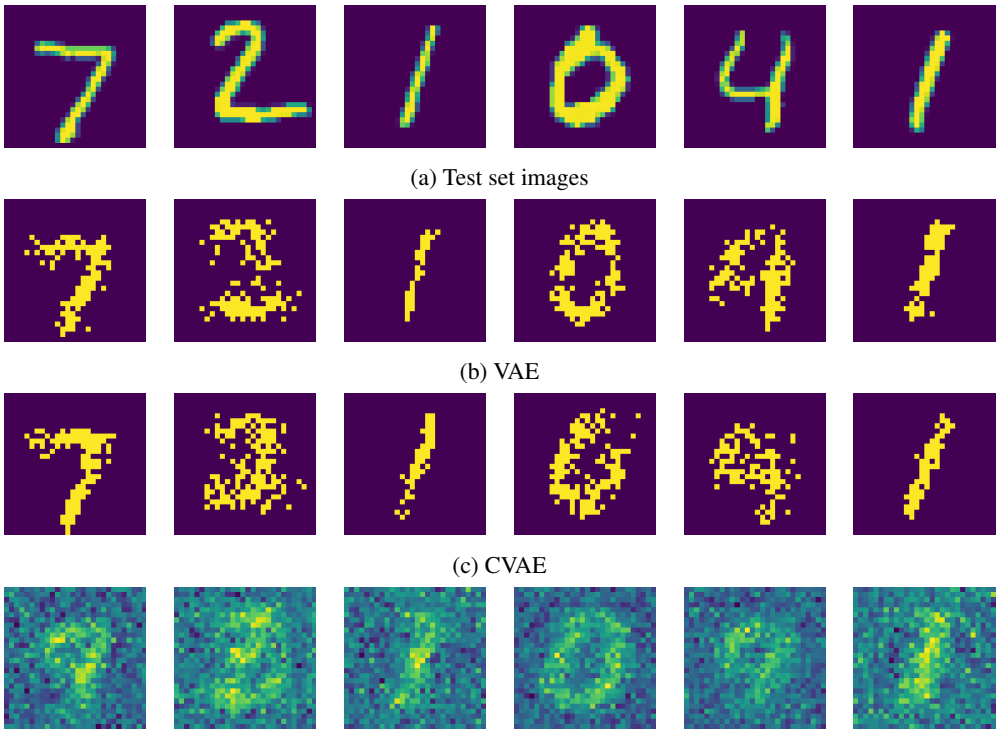
(a) Test set images

(b) VAE

(c) CVAE

(d) PPCA

Figure 1: Comparison of MNIST test set images and corresponding mean parameters generated by density models



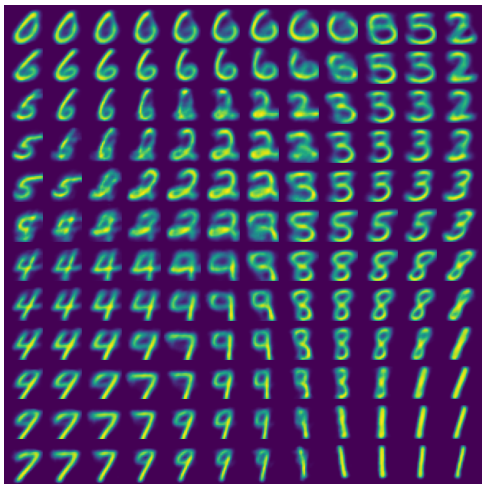
(a) Test set images

(b) VAE

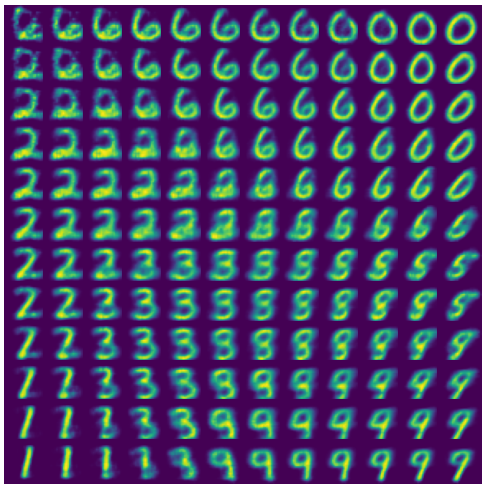
(c) CVAE

(d) PPCA

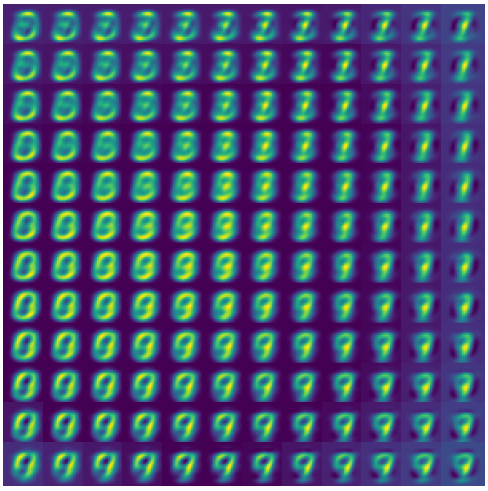
Figure 2: Comparison of MNIST test set images and corresponding reconstructions sampled from density models



(a) VAE



(b) CVAE

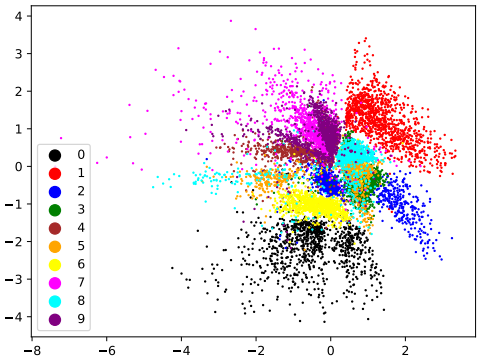


(c) PPCA

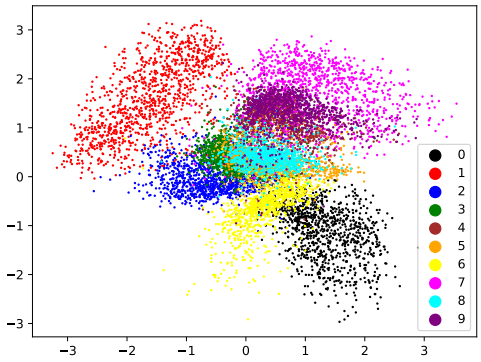
Figure 3: Interpolating images from latent space variables using trained density models.

	Log-Likelihood/ELBO	MSE
VAE	-1.428134e+02	0.037524
CVAE	-1.568176e+02	0.044254
PPCA	-4.329656e+03	3629.250732
GMM	-1.067011e+07	3831.718461

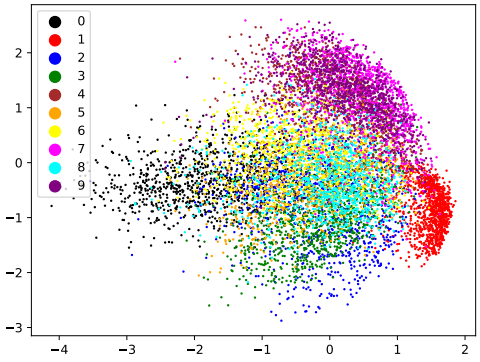
Table 1: Model performance metrics



(a) VAE



(b) CVAE



(c) PPCA

Figure 4: Clustering on MNIST test (projection to latent space) using trained density models.

2 Function fitting

2.1 Fitting a GP with Pyro

2.2 Bayesian Optimization

3 Bibliography