Comparison of different autoencoder architectures with applications

Ben Gerhards[0009−0002−4035−6696]

Intelligent Embedded Systems, University of Kassel, 34109 Kassel, Hessen, Germany

Abstract. WIP

Keywords: Variational autoencoder, Variational Inference, VAE

1 Introduction

Variational Autoencoders (VAEs) have been introduced by Kingma et al in 2013 [14]. They are an extention of autoencoders which are composed of two parts an encoder and a decoder. The decoder maps data into a lower dimensional space and the decoder takes this lower dimensional representation and decodes it back into the original space, trying to reconstruct the original input. The difference between the autoencoder and the VAE is the modeling of the latent space. The autoencoder encodes a sample into a point in the latent space, while the autoencoder encodes it into a distribution over the latent space. This extention is achieved by variational inference. The first pratical application of the VAE by Kingma et al. was to generate black and white images of faces and handwritten digits. Since they showed that their approach works with image data VAEs where in the beginning most porminently used as generative networks and applied to the task of image generation. But with more time and research VAEs got applied to a wider variety of problems with other applications than just as a generative network. In this paper we will summarize multiple papers where VAEs where applied to different fields of machine learning. The goal is to study what makes VAEs useful or what adaptations are neccessary for different problems and how the performance of VAEs can be improved.
The rest of the paper is structured as follows. First autoencoders and their extention to variational autoencoders are explained and a qualitative comparision to other generative models are made in section 2.1.After that different applications for VAEs are discussed with examples and the architectures used. The first application that is discussed is the application to pictures in section 3. Next examples of VAEs used in anomaly detection are given in section 4. In section 5 VAEs are used to augment data. Another big field where VAEs are deployed is that of natural language processing (section 6). Next, in section 7 VAEs are used on time series data. Lastly combinations of the VAE with other architectures are discussed in section 8 and the paper is concluded in section 9.

2 Foundation

This section will cover the idea and originally proposed architecture of the VAE. First the idea behind autoencoders and their architecture is explained. After that, extension from the auto encoder to the variational auto encoder is explained and lastly a quantitive comparison to other generative models will be made.

2.1 Autoencoders

Autoencoders were originaly designed to achieve dimensionality reduction or feature learning for some input [7]. Application areas are for example denoising, where the autoencoder is given data with noise, with the goal to learn a representation of that data that is noise-free.
The autoencoder consits of two parts, the encoder E and the decoder D. The encoder maps the input into the so called latent space and the maps from the latent back to the input space. The objective of the autoencoder is to reconstruct the given input i.e. it should learn the function $\mathbf{x} = D(E(x))$. Since this objective is trivial in general, the autoencoder has to be restricted to be useful. The most commong way to restrict the autoencoder from just learning to copy the input to the ouput is to adapt the architecture. If the encoder is forced to map into a smaller space than that of the input, the model is forced to learn a compressed representation of the input and is not able to learn the identity anymore. This approach is called an undercomplete autoencoder. In this basic case the objective function to be optimized is the mean-squared-error (MSE):

$$L(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - D(E(\mathbf{x}))\|^2 \tag{1}$$

Another example is the so-called sparse autoencoder. This time, instead of forcing the decoder to use a smaller space to map into, we restrict the output of the decoder to be very small in every dimension. To achieve this a regularization term is added to the objective function. The objective function of the sparse autoencoder then becomes

$$L(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - D(E(\mathbf{x}))\|^2 + \lambda \sum_{i=1}^{M} |E(x)_i| \tag{2}$$

where λ is a hyperparameter to weight the regularization, M is the dimensionality of the latent space.

2.2 Variational Autoencoders

Variational auto encoders are another form of regularized auto encoders. They were introduced by Kingma et. al. introduced variational autoencoders in [14]. As the name implies, the structure of the new architecture remains the same but uses variational inference to regularize the latent space to force the autoencoder

to learn a usefull representation. Instead of mapping the inputs to a singular point in the encoderspace the encoder of the VAE encodes the data as a distribution over the latent space. This way the latent space can be regularized by enforcing it to be close to a prior distribution that is chosen beforehand and has a usefull shape. In practice this prior distribution is often chosen to be a multivariate standard normal distribution $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2\mathbf{I})$. The encoder then returns the latent distribution $p(\mathbf{z}|\mathbf{x})$ in the form of the mean $\boldsymbol{\mu} \in \mathbb{R}^M$ and the covariance matrix $\boldsymbol{\sigma}^2\mathbf{I}$, with $\boldsymbol{\sigma} \in \mathbb{R}^M$ and M being the dimensionality of the latent space. The decoder then takes a sample $\mathbf{z} \sim p(\mathbf{z}|\mathbf{x})$ from the latent distribution and tries to decode that into a recreation of the input.
Without changing the original loss function the VAE could learn to ignore this restriction and the results would be the same as with the autoencoder. This could be achieved by the model if the encoder learns to output very sharp distributions, where $\boldsymbol{\sigma} \approx \mathbf{0}$. To actually regularize the latent space a regularisation term is added to the original loss function of the autoencoder. This regularization is achieved with the Kullback-Leibler-Divergence (KL-Divergence) between the returned distribution and the prior. The loss function then becomes

$$L(\mathbf{x}) = \|x - D(E(\mathbf{x}))\|^2 + D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I}), \mathcal{N}(\mathbf{0}, \mathbf{I})) \tag{3}$$

The KL-Divergence is never negative and if the distributions are more different the KL-Divergence will be larger. In the case of a multivariate normal distribution and a standard normal distribution, as in our case, the KL-Divergence can be rewriten and the loss function becomes:

$$L(\mathbf{x}) = \|x - D(E(\mathbf{x}))\|^2 + \frac{1}{2}\sum_{i=1}^{D}(\boldsymbol{\sigma}_i^2 + \boldsymbol{\mu}_i^2 - 1 - ln(\boldsymbol{\sigma}_i^2)) \tag{4}$$

This approach holds another problem that needs to be solved. By sampling a random variable from the distribution returned by the encoder, it is not possible to backpropagate the error from the loss function through the model, as the sampling step is not differentiable. THis makes it impossible to train the model with gradient based methods. As a solution they introduce the reparametrization trick [14]. Instead of sampling from the returned distribution directly, they sample another random variable $\boldsymbol{\epsilon}$ from a multivariate standard normal distribution and reparametrize $\mathbf{z}$ as $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}^2\mathbf{I} \odot \boldsymbol{\epsilon}$. $\odot$ indicates an elementwise multiplication. This way $\boldsymbol{\epsilon}$ can be viewed as another input and $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ can be viewed as additional parameters and the gradient can be backpropagated again. With this trick the whole framework is trainable entirely with gradient based methods.

2.3 Quantitative Comparision To Other Generative Models

One popular way to use VAEs is as a generative model. Since the prior distribution is chosen by the user, the trained decoder can be used to generate data by sampling from that prior and feeding that input to the decoder.

The other two popular models for generating data are generative adversarial networks (GANs) [8] and denoising diffusion probabilistic models (DDPMs) [10]. Since these three architectures form the group of the most popular generative models, we will make a short quantitative comparison between them.
The first mayor difference between these models is their way of generating data after training. All three models use normaly distributed noise and transform it to generate new data but compared to GANs and DDPMs, VAEs have the advatage of an interpretable latent space. For example the latent space of the VAE can be examined to find those dimensions that hold meaningful information. These dimensions show a larger KL-Divergence between itself and the prior than the others. Such interpretations are not possible with GANs and DDPMs as their latent space is not ragularized and not interpretable. When we compare the generated data of the three models it becomes obvious that the data generated by the VAE is in general of less quality and often lacks high frequency information [16].
One downside of DDPMs is their long inference time, meaning compared to GANs and VAEs it takes a long time to generate data from noise [3]. On the other hand DDPMs generaly don't exhibit problems like mode or posterior collapse as is common in GANs and VAEs.
In general the training of VAEs is not as straight forward and stable as thetraining of DDPMs and GANs as the right value for the regualrization weight of the KL-Divergance is critical. The other architectures do not have such a hyperparameter that that is that significant for the training.
There are adaptations of the basemodels to mitigate the downsides of these models like for example using the Wasserstein distance in the training of GANs which reduces modecollapse but most of these problems are inherent. These differences lead to different models beeing suitable for different problems.

3 Image Generation and Reconstruction

One very prominent application of VAEs is the generation and manipulation of images like inpainting or alteration. In general, image generation is performed by training the VAE in an unsupervised fashion to reconstruct the given images. After training, only the decoder part is kept and to generate new images a sample is drawn from the prior distrbution and fed through the decoder to get a new image. One simple way to perform image reconstruction or inpainting is to encode the incomplete image and to decode that to get a complete image. Since the VAE has an meaningfull latent space image alteration can be done directly in that latentspace by vector arithmetic for example.
Yan et al. used VAEs for generating images, as well as their reconstruction and completion from prompts formulated in natural language [23]. In their model, disCVAE, is an extention of the VAE to a conditional model. Their approach is unique in that sense that they composite the image from a background and a foreground. The background and foreground use their own VAEs and are additionaly conditioned on the attributes $\mathbf{y}$ extracted from the prompt with the

help of another model. In the end the forground and background get merged by a mask that is also output by the VAE that produces the foreground. To implement the VAE they use CNNs and fully connected layers for all neural networks. To generate new images latent variables for both latentspaces are sampled, the prompt is encoded into a vector using the same model as before, these are given to the VAEs as inputs and the outputs are combined to get the final image. For image reconstruction a fitting latent variabel that can be decoded has to be found. This is done by finding the the latent variabel z that maximizes $p_\theta(z|x_0, y)$ where x_0 is the incomplete image. They show that this can be done by solving an energy minimization problem. This latent variable can then be decoded by the decoder to get a reconstructed image.

WIP:
[17] Use labels and captions for semi-supervised learning of image generation. Convolutional layers stochastic unpooling together with Bayesian SVM

Another example of a VAE that is used for image generation is the NVAE from Vahdat and Kautz [21]. Their goal is to create VAEs that can generate high qualtiy images with a high resolution. Instead of the traditional VAE they used a hierarchical VAE for this [18]. To achieve very deep hierarchical VAEs they introduce residual cells with depthwise convolutional layers. These cells use skip connections similar to those used in ResNet, that sum up the input and the output of the cell. These connections make it easier for the model to learn long range dependencies by simplifying learning the identity function. To stabilize training they use batch normalization. To further boost performance they use the swish activation $f(u) = \frac{u}{1+e^{-u}}$ instead of the commonly use ReLU or ELU and a channel wise gating layer called Sequeeze and Excitation. They introduce two ways to stabilize the training of deep hierarchical VAEs namely spectral regularization of the residual normal distributions and the addition of normalizing flows into the modelling of the distributions. Lastly they also performed a ablation study to validate their claims. These studys show that the proposed additions and innovations do increase the performance and stability of the training of the VAE.

WIP:
[19] Generating Diverse High-Fidelity Images with VQ-VAE-2 Fully connected layers in de and encoders. multi-scale hierarchical organization of VQ-VAE, augmented with powerful priors over the latent codes, non-linear mapping into latent space -¿ VQ -¿ descrete latent variables -¿ indicies -¿ Vectors -¿ decoder, shared codebook, codebook-loss commitment-loss Use a hierarchy of vector quantized codes to model large images The main motivation behind this is to model local information, such as texture, separately from global information such as shape and geometry of objects.

4 Anomaly Detection

The way VAEs work make them a good fit for anomaly detection, which is used in a variety of fields.

One way to perform anomaly detection is to use the reconstruction error. In this case the VAE is trained and new samples are classified by feeding the sample through the encoder and decoder. After that the reconstruction error is measured and if that exceeds a certain threshold the sample is classified as an anomaly. In contrast to that An and Cho use reconstruction probability instead of reconstruction error [2]. They use a very simple VAE, where the decoder and encoder are a single fully-connected layer with 400 hidden units and the latent space being 200 dimensional. Instead of outputting a single value the decoder in this case outputs the parameters of a multivariate isotropic gaussian. To classify whether or not a new sample is anomolous the sample is encoded with the encoder and multiple samples from the output latent space are drawn. These latent variables are decoded into a sample distribution. After that, the reconstruction probability is calculated as the average probability of the sample under each of the output sample distributions. If this probability is smaller than a threshold the sample is classified as an anomaly.

WIP:
[15] LSTM-Based VAE-GAN for Time-Series Anomaly Detection Encoder, Generator, Disctiminator. anomalies are detected based on reconstruction difference and discrimination results Industrial time series data. Reconstruction approach (vs prediction approach)

5 Data Augmentation

The task of data augmentation is to generate additional data that is similar to provided data. With data augmentation it is possible to extend datasets to provide more training data for other tasks making it possible to achieve better generalization. For this to work the samples generated do have to be both similar enought for the data to be usefull but also novel enough so that the additional data provides additional information. The ability to generate new, artificial data with VAEs makes it possible to be used for dataaugmentation.

To improve the performance of speech recognition systems Hsu et al. used VAEs to generate novel speech data in [11]. Their goal is to generate samples that share the same transcript as a sample from the given dataset but differ in other features like speaker identity or channel and background noise. They utilize a sequence to sequence (Seq2Seq) recurrent VAE that uses a two layer LSTM to model the encoder and decoder. The encoder encodes the given speech features by using the ouput from both LSTM-layers as input to the gaussian parameter layer that outputs the parameters for the latent distribution. The latent representation is then decoded by the decoder by using z as input for the same amount of time steps as the original input is long. The output of the last LSTM-layer is used again as input to a gaussian parameter layer that outputs a

distribution from which can be sampled to get the reconstruction of the input. To augment the data they change nuisance attributes of the speech meaning not the linguistic content but only the surface form of the speech. They present two ways to modify the original samples namely nuisance attribute replacement and atent nuisance subspace perturbation. By assuming the that the VAE uses orthogonal subspaces to model phone and nuisance attributes and that the nuisance attributes are consistent withing an utterance they first compute one latent nuisance representation μ_{utt_j} for each utterance. This done by averaging the means of the latent representation of all frames of the utterance. In nuisance attribute replacement this latent nuisance representatin is replaced by another one via $\tilde{\mathbf{z}}^{(i)}_{utt_{src}} = \mathbf{z}^{(i)}_{utt_{src}} - \mu_{utt_{src}} + \mu_{utt_{tar}}$. $\tilde{\mathbf{z}}^{(i)}_{utt_{src}}$ is then decoded to get the altered sample.
In latent nuisance subspace perturbation the subspace is examined to find the factors that have a high impact on the nuisances. To achieve this, μ_{utt_j} i calculated for every utterance in the dataset. Next PCA is performed on this set of representations is performed to get the directions that have the highest variance. In hard latent nuisance subspace perturbation pertubation is only performed in the first N directions with the highest variance. In constrast to that, in sthe soft variant a pertubation vector is calculated as $\mathbf{p} = \gamma \sum_{d=1}^{D} \psi_d \sigma_d \mathbf{e}_d, \psi_d \sim \mathcal{N}(0,1)$ where σ_d is the square-root of the d-largest eigenvalue and $\mathbf{e}_d$ its eignevector and γ is a hyperparameter. In $\mathbf{p}$ the pertubation along each direction is proportional to the vairance in that direction and $\tilde{\mathbf{z}}^{(i)}_{utt_{src}}$ is then calculated as $\tilde{\mathbf{z}}^{(i)}_{utt_{src}} = \mathbf{z}^{(i)}_{utt_{src}} + \mathbf{p}$.

WIP:
[12] Data augmentation for time series regression deal with imbalanced data fully connected layers adjusted by a Bayesian optimised parameter additionally, an L2 penalty, weighted by a Bayesian optimised parameter , is added.

WIP
[4] Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting Compare Autoencoders, VAEs and WGAN-GPs for day-ahead electricity prices. Autoencoders with crossentropy and not mse, weighted D_{KL} regularizer and L2-penalty sampling is again performed by samppling from the prior and the samples is fed to the decoder. applied to electricity prive forecasting In their evaulation VAEs perform very good

6 Natural Language Processing

Natural language processing is a wide field that deals with the problem of enabling computers to understand human language in form of text or speech. It includes tasks like transforming text to speech, recoginizing speech and generating and classifying text.

The first paper we look at is [1]. In this paper Akuzawa et al. combine VoiceLoop, an autoregressive model, with a VAE to enable VoiceLoop to create expressive speech taht is also of higher quality. Their proposed model is a

VAE that consists of a convolutional neural network as encoder and then uses voiceloop as a decoder. The idea is, that by providing auxiliary information c in form of the phonemes,to the encoder, only additional information such as expression is encoded in the latent space. By then using the latent variable as well as the phonemes in VoiceLoop, the created speech should more natural. Training is then similar to the basic training of VAEs where the reconstruction error is minimized and the KL-Divergence is used as regularization. To encourage VoiceLoop to use the latent variable they also use KL cost annealing, where the weight of the KL-Divergence is adapted over time. At the start of the training they set the weight to 0, so that the model encodes as much information as possible in the latent space and then increase it to one during the course of training.

WIP:
[20] A Hybrid Convolutional Variational Autoencoder for Text Generation the core difficulty of training VAE models is the collapse of the latent loss (represented by the KL divergence term) to zero. propose a novel VAE model for texts that is more effective at forcing the decoder to make use of latent vectors. Use CNN layers with RNN language model layer architecture where the feed-forward part is composed of a fully convolutional encoder and a decoder that combines deconvolutional layers and a conventional RNN. Two training tricks are required to mitigate this issue: (i) KL-term annealing where its weight in Eq (1) gradually increases from 0 to 1 during the training; and (ii) applying dropout to the inputs of the decoder to limit its expressiveness and thereby forcing the model to rely more on the latent variables. in addition to the reconstruction cost computed on the outputs of the recurrent language model, we also add an auxiliary reconstruction term computed from the activations of the last deconvolutional layer:

WIP:
[22] Variational Autoencoder for Semi-Supervised Text Classification Semi-supervised Sequential Variational Autoencoder (SSVAE) is proposed, which increases the capability by feeding label into its decoder RNN at each time-step. Solution: feed the label at every timestep instead of only the first one. Motivate their approach by looking at the problem from a reinforcment point of view that inspects the gradient of the classifier used for semi-supervised learning. The first one concatenates word embedding and label vector at each time-step Second: directly input the label into the memory cell, not through gates. Use sampling to reduce the computational complexity. Works.

WIP:
[13] Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech apply normalizing flows to our conditional prior distribution and adversarial training on the waveform domain With the uncertainty modeling over latent variables and the stochastic duration predictor, our method captures speech variations that cannot be represented by text. Adversarial training of the decoder. additional Losses (a lot). The overall architecture of the proposed model consists of a posterior encoder, prior encoder, decoder, discriminator, and stochastic duration predictor.

7 Time Series Generation

VAEs can not only be used to generate images i.e. 2D-data but also time series i.e. 1D-data by applying the same principles as in image generation.

TimeVAE, propsed by Desai et al. in [5] is a VAE that is specifically designed for generating time series. Their base model, TimeVAE, is a VAE that uses convolutions and a dense layer to map into and out of the latent space. They then extend the base model to the Interpretable TimeVAE by introducing custom blocks for level, trend and seasonality in the decoder to enable the model to use user-defined distributions. This makes their approach more interpretable has the advatange that specific domain knowledge can be used. In the trend block the user can define the number of degrees that the polinomial that models the trend should have. They then use the latent vector to predict the basis expansion coefficients. The trend is then constructed by $\mathbf{V}_{tr} = \theta_{tr}\mathbf{R}$ where $R = [\mathbf{1}, \mathbf{r}, \ldots, \mathbf{r}^p]$ with $\mathbf{r} = [0, 1, \ldots, T-1]^T/T$. The level is then just the level of $p = 0$. The decoder enablesthe use of an arbitrary number of seasonalities to be modeled by using multiple seasonality blocks. Each block has two parameters, the number of seasons m and the duration of each season d. Again the coefficients are estimated from the latent variable z, this time for each timestep independendly, then indexed and then summed up to get the seasonality estimates.
Interpretability is achieved by explicitly predicting the base coefficients for the blocks. The final output of the decoder is then the elementwise sum of all the blocks that are used and the base decoder. It is also noteworthy that the basedecoder as well as all of the blocks in the decoder are implemented as residual branches meaning, that they can be enabled or disabled individually. They also introduce another hyperparemter in training to put more or less emphasis to the KL-Divergence.
They apply their model on stockmarket as well as energy and air data to show that their model works well with different kinds of time series data.

WIP:
[6] The challenge of realistic music generation: modelling raw audio at scale autoregressive discrete autoencoder and argmax autoencoder enlarge receptive field of ar models. we turn an AR model into an autoencoder by attaching an encoder to learn a high-level conditioning signal directly from the data. So encoder -¿ representation -¿ AR model Use Autoencoders with AR because regularisation leads to posterior collapse. Instantiations are then VQ-VAE and AMAE. VQ-¿ Codebookcollapse -¿PBS -¿ expansive -¿ AMAE. Quantisation without codebook. Encoder produces k-dimensional queries, and features a nonlinearity that ensures all outputs are on the (k-1)-simplex. Add diversity loss to encourage the model to use all outputs in equal measure. AMAE underperforms when having the same architecture but has better convergence they use wavenet for all subnetworks and average pooling at the output.

WIP:
[9] A variational autoencoder for music generation controlled by tonal tension Controllable music generation system in terms of tonal tension. Measures used are the cloud diameter and tensile strain from Spirtal Array theory. Instead of

just reconstructing x in form of a feature set the decoder also outputs the tensile strain and cloud diameter. They add an additional loss in form of the MSE between the output and the original. Gated recurrent units as de and encoder. Then manipulate latent space to produce another tension. They found two vectors in the latent space. With them it is possible to change the tension/cloud diameter direction and tension/cloud diameter level. Done so by labeling the data by performing correlation analysis.

8 Hybrid Networks

As described in section 2.3 different generative models do have different strength and weaknesses. This motivates the combination of different architectures to mittigate some of the downsides and create even more powerfull models. One such model was already mentioned in section 4 in form of the VAE-GAN [15].
Another example is the DiffuseVAE that combines a DDPM with a VAE. It was proposed by Pandey et al. in [16] for large image generation and denoising. The idea behind their model is to generate noisy images of less quality with a VAE and refine them afterwards with the help of the DDPM. This way the denoising time can be drasticaly reduced, while still generating samples of high quality. Additionaly the proposed model has all the upsides of a VAE meaning that it has an interpretable latent space that can be used to control the generation of the images. With some simplifying design choices they show that the training of their propsed model boils down to a two stage training where in the first stage the VAE is trainedon a sample x_0 and in the second stage the DDPM is trained and conditioned on the output of the VAE. In the paper they use a simple VAE and two different parameterizations for the DDPM. One where the forward process is independent of the output of the VAE but the reverse process is and one where both are conditioned on the output of the VAE. The second variant proves to generate samples of a little less quality judged by the FID score. They also examine the effects of interpolation in the VAE and the DDPM latent space. This investigatino shows that the VAE latent space encodes major image structure like face orientation, gender, facial expressions, while the DDPM latent space only encodes minor features like skin tone or lip color. Next they show that the synthesis of images can be controlled via vector arithmetic in the VAE latentspace like in basic VAEs. The last result from the paper is that the model can handle the denoising of different types of noisy images, not only those generated by the VAE.

9 Conclusion

As a conclustion it can be said that a lot of research on VAEs has been done since their introduction in 2013 to improve them. Their architecture makes them usefull for a wide variety of tasks at which they were applied successfully. The reason VAEs are so versatile is their usefull latent space than can be used in a variety of ways. As the first pratical application of VAEs was image generation

and alteration, this field recieved a lot of attention since their introduction. In this case the latent space is useful as it is close to the chosen prior meaning that it can be sampled from. With these samples new images can be generated by the decoder. Since the latent space is regular and exploitable generation can be influenced by altering the latent variable of images by simple vector arithmetic to generate altered images. Similarly the generation of new images can also be influenced via the latent variable that will be decoded.
VAEs where also successfully applied to the task of anomaly detection. Here again the probabilistic nature of the model as well as the regularized latent space of VAEs make them usefull. Here we saw two approaches, one based of off the reconstruction error and the other one based on the reconstruction probability.
Next we looked at VAEs that were used for data augmentation. Here similarly to the generation of images, new data is generated by sampling from the prior of the latent space to create novel data samples. These samples are then both similar and novel enough to benefit downstream tasks like the training of other models on a larger dataset thus improving generalization and performance.
VAEs can also be applied to tasks of natural language processing like speech synthesis and speech synthesis and classification. Here we saw that VAEs can also be used as classifiers.
In section 7 we saw that VAEs are not only applicable to two dimensional data in the form of images but also for time series. Time series include many types of sequential data like stock market data or music. The summarized papers showed that VAEs could be applied to this kind of data as well by appliying the similar prinicples as before.
Lastly the combination of other generative models in the form of GANs and DDMPs with VAEs to mittigate their respective downsides and improve performance was investigated by us. When combined with DDPMs the quality of generated images can be improved and at the same time the time it takes to generate images of such high quality can be reduced. On the other side when combined with GANs.

References

1. Akuzawa, K., Iwasawa, Y., Matsuo, Y.: Expressive speech synthesis via modeling expressions with variational autoencoder. arXiv preprint arXiv:1804.02135 (2018)
2. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special lecture on IE **2**(1), 1–18 (2015)
3. Bond-Taylor, S., Leach, A., Long, Y., Willcocks, C.G.: Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. IEEE transactions on pattern analysis and machine intelligence **44**(11), 7327–7347 (2021)
4. Demir, S., Mincev, K., Kok, K., Paterakis, N.G.: Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting. Applied Energy **304**, 117695 (2021)

5. Desai, A., Freeman, C., Wang, Z., Beaver, I.: Timevae: A variational auto-encoder for multivariate time series generation. arXiv preprint arXiv:2111.08095 (2021)
6. Dieleman, S., Van Den Oord, A., Simonyan, K.: The challenge of realistic music generation: modelling raw audio at scale. Advances in neural information processing systems **31** (2018)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
8. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014). https://doi.org/10.48550/ARXIV.1406.2661, https://arxiv.org/abs/1406.2661
9. Guo, R., Simpson, I., Magnusson, T., Kiefer, C., Herremans, D.: A variational autoencoder for music generation controlled by tonal tension. arXiv preprint arXiv:2010.06230 (2020)
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
11. Hsu, W.N., Zhang, Y., Glass, J.: Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In: 2017 IEEE automatic speech recognition and understanding workshop (ASRU). pp. 16–23. IEEE (2017)
12. Islam, Z., Abdel-Aty, M., Cai, Q., Yuan, J.: Crash data augmentation using variational autoencoder. Accident Analysis & Prevention **151**, 105950 (2021)
13. Kim, J., Kong, J., Son, J.: Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In: International Conference on Machine Learning. pp. 5530–5540. PMLR (2021)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
15. Niu, Z., Yu, K., Wu, X.: Lstm-based vae-gan for time-series anomaly detection. Sensors **20**(13), 3738 (2020)
16. Pandey, K., Mukherjee, A., Rai, P., Kumar, A.: Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. arXiv preprint arXiv:2201.00308 (2022)
17. Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., Carin, L.: Variational autoencoder for deep learning of images, labels and captions. Advances in neural information processing systems **29** (2016)
18. Ranganath, R., Tran, D., Blei, D.: Hierarchical variational models. In: International conference on machine learning. pp. 324–333. PMLR (2016)
19. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. Advances in neural information processing systems **32** (2019)
20. Semeniuta, S., Severyn, A., Barth, E.: A hybrid convolutional variational autoencoder for text generation. arXiv preprint arXiv:1702.02390 (2017)
21. Vahdat, A., Kautz, J.: Nvae: A deep hierarchical variational autoencoder. Advances in neural information processing systems **33**, 19667–19679 (2020)
22. Xu, W., Sun, H., Deng, C., Tan, Y.: Variational autoencoder for semi-supervised text classification. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
23. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 776–791. Springer (2016)