UNDER REVIEW

RESCIENCE C

Replication / Machine Learning

[Re] Variational Sparse Coding

Alfredo De la Fuente[1,2] and Robert Aduviri[3]

[1]Skolkovo Institute of Science and Technology, Moscow, Russia – [2]National Research University Higher School of Economics, Moscow, Russia – [3]Pontifical Catholic University of Peru, Lima, Peru

Edited by
Koustuv Sinha

Reviewed by
Anonymous
Anonymous

Received
01 May 2019

Published
–

DOI
–

1 Introduction

The paper[1] proposes an improvement over the Variational Auto-Encoder (VAE) architecture[2,3] by explicitly modelling sparsity in the latent space with a Spike and Slab prior distribution and drawing ideas from sparse coding theory. The main motivation behind their work lies in the ability to infer truly sparse representations from generally intractable non-linear probabilistic models, simultaneously addressing the problem of lack of interpretability of latent features. Moreover, the proposed model improves the classification accuracy using the low-dimensional representations obtained, and significantly adds robustness while varying the dimensionality of latent space.

The authors of the paper derive an analytic expression for the evidence lower bound (ELBO) of the VAE model by choosing the sparsity-inducing Spike and Slab prior distribution for the latent variables, which is later optimized using standard gradient methods to recover the encoding and decoding mappings. After training on well-known datasets, the Variational Sparse Coding (VSC) model is able to recover sparse, informative and interpretable representations given a fixed number of latent dimensions, which authors claim to be advantageous over standard VAE representations for classification tasks.

In this reproducibility report we study in detail the VSC model to implement the architectures described in the paper, run the experiments (detailed in Section 4), provide insights and suggestions for replicating results, and analyze the results obtained in comparison with the ones reported by the authors of the paper (Section 6). Furthermore, we add modifications to improve the model performance and propose some possible future work directions (Section 7).

2 Related Work

Variational Auto-Encoders have been extensively studied[4] and widely modified in the recent years in order to encourage certain behavior of the latent space variables[5,6,7] or to be further applied for particular tasks[8,9,10,11]. Regarding the sparsity of the latent space for VAEs, previous work in the literature has focused on either explicitly incorporating a regularization term to benefit sparsity[12], or fixing a prior distribution, such as Rectified Gaussians by[13], discrete distributions by[14], student-t distribution for Variational Information Bottleneck by[15] and Stick Breaking Processes by[5].

Correspondence should be addressed to Robert Aduviri (robert.aduviri@pucp.edu.pe)
The authors have declared that no competing interests exists.
Code is available at https://github.com/Alfo5123/Variational-Sparse-Coding.

Nonetheless, previous works have not allowed to explicitly model sparsity by incorporating linear sparse coding to non-linear probabilistic generative models. The paper we aim to reproduce offers a connection between both areas through the Spike-and-Slab distribution, chosen as a prior distribution for the latent variables. Although this distribution has been commonly used for modeling sparsity[16], it has rarely been applied to generative models. Moreover, since sparse coding imposes efficient data representations[17,18,19], the authors demonstrate qualitatively how the sparse learned representations can capture subjectively understandable sources of variation.

Following the line of latent features interpretability, we can observe that the authors' idea is closely related to the Epitomic VAE by[20], which learns the latent dimensions the recognition function should exploit. Many recent approaches, mostly related to disentangled representations, such as β-VAE[21,22] or Factor-VAE by[23], focus on learning interpretable factorized representations of the independent data generative factors via generative models. However, these approaches although explicitly induce interpretation of the latent features, do not directly produce sparse representations in contrast with the VSC model. Hence, the authors' aim is to develop a model that directly induces sparsity in a continuous latent space, which in addition, results into a higher expectation of interpretability in large latent spaces.

Our work, as part of the reproducibility challenge, will contribute in clarifying the implementation details of the VSC model, corroborate the results, and assess a few concerns of the reviewers by adding small modifications to the model and experiments based on the insights obtained during the reproducibility challenge.

3 Target Questions

In order to assess the reproducibility of the paper and validate its conclusions, the main questions we will focus our efforts on answering are:

- Can we actually validate the reported results?
- Is it possible to interpret the latent features learned by the VSC model?
- How the proposed model can be further improved?

4 Experimental Methodology

Within the experiments described in the paper, we focus on replicating the precise settings for:

- *ELBO evaluation*: to observe ELBO drop while optimizing the model loss.
- *Classification Performance*: to use the learned presentations for classification tasks.
- *Interpretation of sparse codes*: to visually inspect the role of learned latent features.
- *Visualization / Traversal of Latent Space*: to qualitatively evaluate the reconstructed images.

We found that the paper was well written and moderately amenable for reproduction. Although the authors did not make the code available, writing the code from scratch was not as challenging as we anticipated. Thus, in this section, we describe in detail how our implementation of the model was carried out, clarify the adjustments considered and display the results obtained.

4.1 Datasets

We test the VSC model in two commonly used image datasets: MNIST[24] and Fashion-MNIST[25], both composed of 28×28 grey scale images of handwritten digits and pieces of clothing respectively. Following the paper description, we run most of the experiments with these datasets. In addition to this, CelebA faces[26] dataset was used to showcase qualitative results. We include in our repository routines to download and preprocess the datasets.

Observations —

- For the CelebA dataset, we used a subset of 100K samples for training and 20K samples for testing, which were center cropped and downsampled to a size of 32×32 using all 3 RGB channels, as described in the paper.
- We applied a 0-1 Min-Max scaling to all the datasets. We observed that normalizing vectors to unitary norm, as the paper suggests, produced lower quality image reconstruction.

4.2 Implementation Details

We decided to replicate the architecture described in the paper using PyTorch[27]. Our repository includes a few instructions on how to install and set up all the required libraries needed for running our implementation. We organize the code on scripts for each model architecture, as well as Jupyter Notebooks for preprocessing, running experiments and visualization.

In order to establish a valid benchmark, we implemented the VAE architecture from[3], which, in the same way as the Variational Sparse Coding (VSC) model, was implemented by an encoder and a decoder function, both parametrized as fully connected neural networks. The architecture, implemented from scratch, allows to explicitly define the hyperparameters of the model (e.g., hidden layer dimension, latent space dimension, learning rate, epochs, batch size, etc.) and is highly modular, to encourage future modifications.

For the loss function, the authors used a continuous relaxation for the discrete binary component in the reconstruction term of the ELBO and applied the reparametrization trick[28] for the Spike and Slab distribution to obtain a differentiable expression which can be optimized. Implementing this code was straightforward, our implementation forces the parameter c to increase by 0.001 per iteration to benefit convergence stability.

We stored all the checkpoints for the trained models, for reproducibility purposes, together with the training logs which can be visualized using TensorBoard.

Observations —

- One of the missing details in the paper was the batch size. We assumed it to be 32 samples per batch, due to our memory restrictions.
- The original paper suggests using $20,000$ iterations for model training using the ADAM optimizer[29] with learning rate ranging between 0.001 and 0.01. In particular, we implemented the VSC model in a way that the number of epochs is one hyperparameter. Thus, we fixed the number of epochs to be equivalent the number of iterations given by the paper; i.e., for MNIST and Fashion-MNIST we trained the VSC model for 11 epochs with a batch size of 32. We fixed a learning rate of 0.001.

- A minor downside of the paper is that the weights initialization method was not specified. We initialized the weights with uniform random variables using the Kaiming initialization method[30] for all the layers, which is also the default method for linear layers in PyTorch.
- For the recognition function, in order to avoid numerical instability, we suggest to either use *clamp* or Sigmoid activation function to avoid spike values of zero (thus ensuring $\gamma_i < 1$).

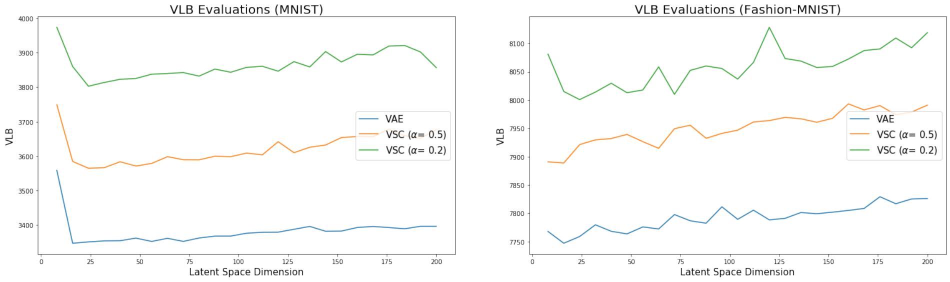
4.3 Reproducibility cost

Given that there was no mention in the paper about computational resources required, we describe the computational cost required for running the experiments. Since MNIST and Fashion-MNIST are relatively small datasets, the training procedure run on CPU took around 1 minute per epoch and 8 seconds per epoch on a Titan Xp GPU, using a latent size of 200 units and a single hidden layer of 400 units for both the encoder and decoder, as described in the paper. On the other hand, the VSC model training time for the CelebA dataset was around 30 seconds per epoch on a Titan Xp GPU, using a latent size of 800 units and two hidden layers of 2000 units also for the encoder and decoder. Regarding memory requirements on GPU, the network trained on MNIST and Fashion-MNIST consumed around 529MB, which scaled up to 850MB during training, while the network trained on CelebA consumed around 637MB, which scaled up to 1322MB during training. In conclusion, the computational cost for running the experiments is not high, which facilitated the reproduction of the results.

5 Results

5.1 ELBO Evaluation

We evaluated how the Variational Lower Bound (VLB) varies while changing the latent space dimension of the models: VSC - $\alpha = 0.2$, VSC - $\alpha = 0.5$ and VAE (Figure 1). We observed that in general the ELBO decreases until reaching an optimal latent dimension and then they slowly increase as we add more latent dimensions.



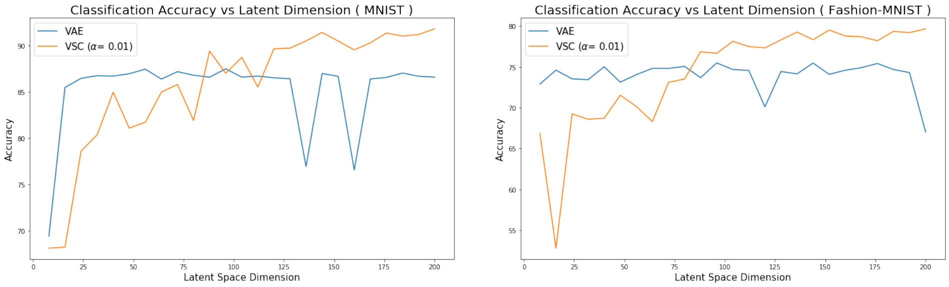
(a) (b)

Figure 1. Evaluation of ELBO on the test set for the VSC trained model at varying number of latent dimensions.

5.2 Classification Performance

We studied the classification accuracy obtained by using the latent representations as input (Figure 2), in order to validate the authors' conclusion that the Variational Sparse

Coding learned representations improve the classification results and significantly increase robustness with respect to the latent number of dimensions. We can observe that for both MNIST and Fashion-MNIST, after surpassing the optimal latent dimension of the VAE, the representations learned by the VSC are significantly more informative as the latent dimension increases.

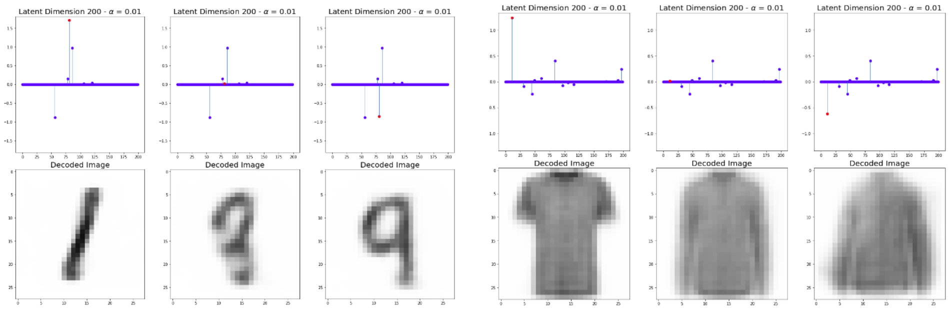


(a) (b)

Figure 2. Classifications results on the test set (for MNIST (a) and Fashion-MNIST (b)) by varying the latent space dimensions using the model VSC - $\alpha = 0.01$ and a 2-layer fully connected neural network as base classifier.

5.3 Intepretation of sparse codes

Authors claim the interpretability of the sparse learned representations. We qualitatively examined the interpretation of the non-zero elements in the sparse codes recovered with the VSC model by running interpolations in the sparse space varying the dimension with the highest absolute value (Figure 3). In addition, we observe the effect of the latent dimensionality at capturing the image content (Figure 4).

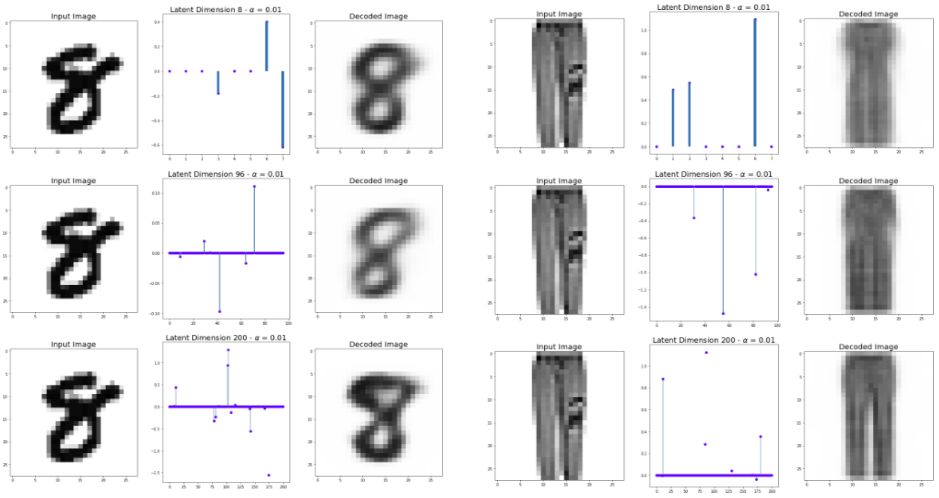


(a) (b)

Figure 3. Reconstruction results by modifying encoding in 200-dimensional latent space for MNIST (a) and Fashion-MNIST (b) for the model VSC - $\alpha = 0.01$.

5.4 Visualization / Traversing Latent Space

We explored how sampling from the latent space distribution can allow us to obtain interpretable variations in the generated images (Figure 5), and also how conditional sampling produces arguably realistic new samples from the same conceptual entity (Figure 6). The traversal of the latent space is performed varying the latent codes with a high absolute value for a given image, one at a time. We can observe that these latent codes indeed represent interpretable features of the datasets, such as the digits shape in MNIST,



(a) (b)

Figure 4. Encoding and reconstruction of samples using the VSC - $\alpha = 0.01$ model from MNIST (a) and Fashion-MNIST (b) datasets with varying latent dimensions of 8, 96 and 200.

the color and shape of the clothes in Fashion-MNIST and the orientation, background color, skin color and hair color in the CelebA results.

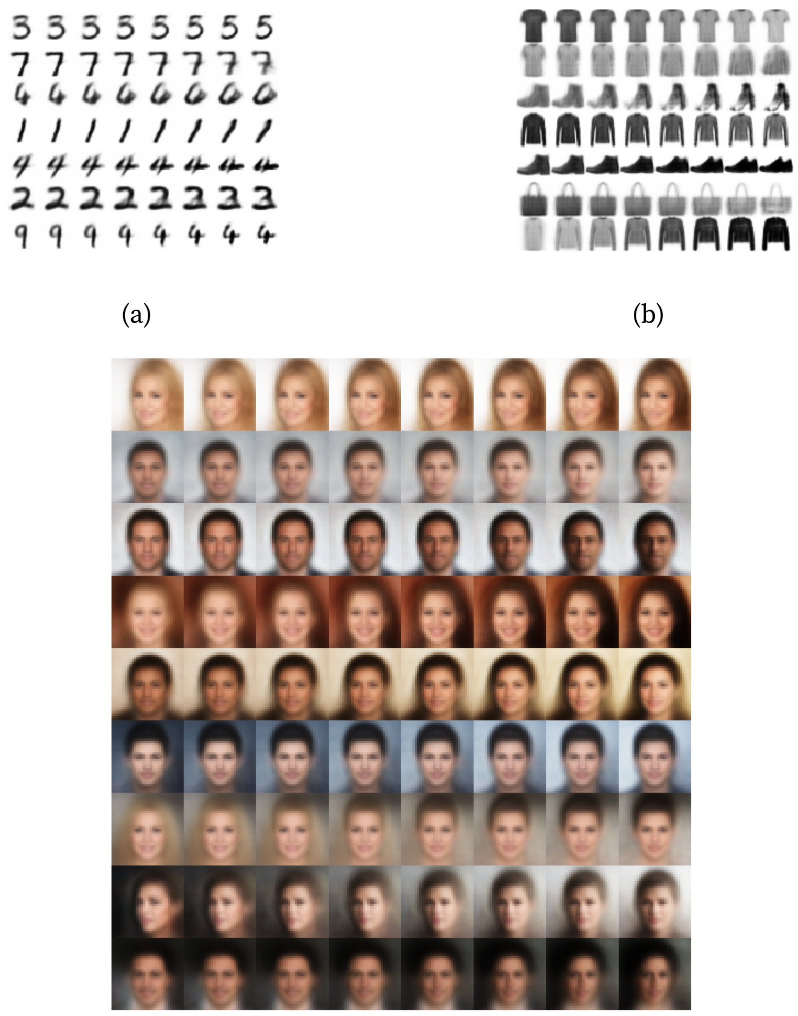
6 Analysis and discussion of findings

We should note that the paper we reproduced was clearly written and overall, despite a few implementation details unspecified (batch size, number of epochs for the CelebA training and weights initialization procedure), it is possible to replicate the results reported. Moreover, we are able to validate the authors hypothesis that the VSC models generate sparse, informative and interpretable representations.

Minor discrepancies in the results can be explained by the restricted number of training epochs. In particular, in Figure 1, we observe a similar trend that the one shown in the paper; however, the gap between the curves can be decreased by training the model for larger number of iterations. Furthermore, in the log optimization files, we can easily notice that by using only $20K$ iterations the model has not reached a local optimum yet. This situation is also present during training on the CelebA dataset: we noticed that at least 50 epochs are needed to be close to a local optimum, as it is shown in Figure 7. We believe that authors must provide more specifications on the optimization hyperparameters to facilitate the reproducibility task.
Similarly, for the classification task (Figure 2) although we were able to obtain more informative representations using the VSC model with respect to VAE, the classification accuracy can still be improved by using a higher capacity classification model, instead of the simple classifier used in the original paper.

Many of the reviewers addressed the issue of how we can interpret the learned latent features by the VSC-model. Although authors affirm that the model does not explicitly induce intepretation, it certainly results into a higher expectation of interpretability in large latent spaces, provided that the sources of variations in the observed data can be considered sparse.

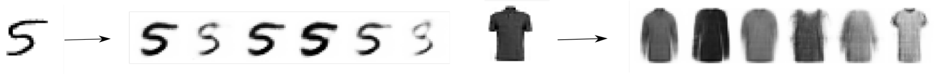
To account for the blurriness of generated images (Figures 3 and 5), we must under-



(a) (b)

(c)

Figure 5. Traversing the latent space of the VSC - $\alpha = 0.01$ model trained for 11 epochs on MNIST (a) and Fashion-MNIST (b) datasets, and 90 epochs on CelebA (c) dataset.



(a) (b)

Figure 6. Conditional sampling from VSC - $\alpha = 0.01$ model trained for 11 epochs on MNIST (a) and Fashion-MNIST (b) datasets.

stand that the model capacity is limited and perhaps convolutional architecture could drastically improve the quality of results.

7 Conclusions

Overall the paper describes in enough detail the VSC model implementation and we must applaud the authors in their ability to convey such a complex topic in an approach-

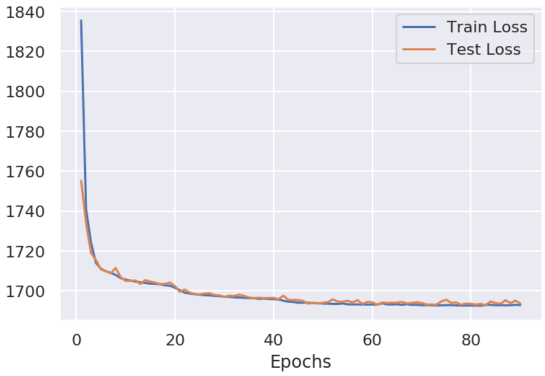


Figure 7. Evaluation of training and test loss of the VSC model on the CelebA dataset

able and replicable manner. The hypothesis of using Variational Sparse Coding to obtain sparse, informative and interpretable representations was confirmed by reproducing the experiments.

Further development on testing the model and comparing it against other sparse models on well known benchmarks is critical. In order to assess how *interpretable* the learned latent features are, we could draw ideas from disentangled representations, to measure the effect of sparsity in the disentanglement metric, against benchmark models such as β-VAE or Factor-VAE.

To conclude our reproducibility report we propose the following directions for future research and improvement of current results:

- Increase the number of iterations suggested for training the models and re-run the experiments using the notebooks in our repository.
- In order to improve the quality of generated images, we suggest to expand the model capacity by either stacking more layers or trying convolutional architectures for the encoder and decoder.
- Apply the model to the Disentanglement testing Sprites dataset, to measure the effect of sparsity on a disentanglement metric.

Acknowledgments — We would like to thank to Emilien Dupont for clarifying distinct aspects on variational autoencoders' implementation, and Lukas Mosser for his suggestions on training generative models.

References

1. F. Tonolini, B. S. Jensen, and R. Murray-Smith. **Variational Sparse Coding**. 2019.
2. D. J. Rezende, S. Mohamed, and D. Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models." In: (Jan. 2014). arXiv: 1401.4082.
3. D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes." In: (Dec. 2013). arXiv: 1312.6114.
4. C. Doersch. "Tutorial on Variational Autoencoders." In: (June 2016). arXiv: 1606.05908.
5. E. Nalisnick and P. Smyth. "Stick-Breaking Variational Autoencoders." In: (May 2016). arXiv: 1605.06197.
6. J. T. Rolfe. "Discrete variational autoencoders." In: **arXiv preprint arXiv:1609.02200** (2016).
7. F. P. Casale, A. Dalca, L. Saglietti, J. Listgarten, and N. Fusi. "Gaussian Process Prior Variational Autoencoders." In: **Advances in Neural Information Processing Systems**. 2018, pp. 10390–10401.

8. X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. "Variational lossy autoencoder." In: **arXiv preprint arXiv:1611.02731** (2016).
9. J. Walker, C. Doersch, A. Gupta, and M. Hebert. "An uncertain future: Forecasting from static images using variational autoencoders." In: **European Conference on Computer Vision**. Springer. 2016, pp. 835–851.
10. M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. "Grammar variational autoencoder." In: **arXiv preprint arXiv:1703.01925** (2017).
11. W. Jin, R. Barzilay, and T. Jaakkola. "Junction Tree Variational Autoencoder for Molecular Graph Generation." In: **arXiv preprint arXiv:1802.04364** (2018).
12. C. Louizos, M. Welling, and D. P. Kingma. "Learning Sparse Neural Networks through L_0 Regularization." In: **arXiv preprint arXiv:1712.01312** (2017).
13. T. Salimans. "A Structured Variational Auto-encoder for Learning Deep Hierarchies of Sparse Features." In: **arXiv preprint arXiv:1602.08734** (2016).
14. A. van den Oord, O. Vinyals, et al. "Neural discrete representation learning." In: **Advances in Neural Information Processing Systems**. 2017, pp. 6306–6315.
15. M. Chalk, O. Marre, and G. Tkacik. "Relevant sparse codes with variational information bottleneck." In: (May 2016). arXiv: 1605.07332.
16. I. J. Goodfellow, A. Courville, and Y. Bengio. "Large-Scale Feature Learning With Spike-and-Slab Sparse Coding." In: **Arxiv** 1 (June 2012), pp. 1–21. arXiv: 1206.6407.
17. H. Ishwaran, J. S. Rao, et al. "Spike and slab variable selection: frequentist and Bayesian strategies." In: **The Annals of Statistics** 33.2 (2005), pp. 730–773.
18. M. K. Titsias and M. Lázaro-Gredilla. "Spike and slab variational inference for multi-task and multiple kernel learning." In: **Advances in neural information processing systems**. 2011, pp. 2339–2347.
19. Y. Bengio, A. Courville, and P. Vincent. "Representation learning: A review and new perspectives." In: **IEEE transactions on pattern analysis and machine intelligence** 35.8 (2013), pp. 1798–1828.
20. S. Yeung, A. Kannan, Y. Dauphin, and L. Fei-Fei. "Epitomic Variational Autoencoders." In: (2016).
21. I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. "beta-vae: Learning basic visual concepts with a constrained variational framework." In: (2016).
22. C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. "Understanding disentangling in β-VAE." In: (Apr. 2018). arXiv: 1804.03599.
23. H. Kim and A. Mnih. "Disentangling by Factorising." In: (Feb. 2018). arXiv: 1802.05983.
24. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: **Proceedings of the IEEE** 86.11 (1998), pp. 2278–2324.
25. H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." In: **arXiv preprint arXiv:1708.07747** (2017).
26. Z. Liu, P. Luo, X. Wang, and X. Tang. "Deep learning face attributes in the wild." In: **Proceedings of the IEEE International Conference on Computer Vision**. 2015, pp. 3730–3738.
27. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. "Automatic differentiation in PyTorch." In: **NIPS-W**. 2017.
28. D. P. Kingma, T. Salimans, and M. Welling. "Variational Dropout and the Local Reparameterization Trick." In: **Advances in Neural Information Processing Systems 28**. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 2575–2583.
29. D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization." In: **arXiv preprint arXiv:1412.6980** (2014).
30. K. He, X. Zhang, S. Ren, and J. Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification." In: **CoRR** abs/1502.01852 (2015). arXiv: 1502.01852.