Category-Learning with Context-Augmented Autoencoder

D. V. Kuzminykh[I], L. P. Kushnareva[II], T. A. Grigoryev[I], A. V. Zatolokin[I]

[I] Moscow Institute of Physics and Technology, Dolgoprudny, Russia
[II] Lomonosov Moscow State University, Moscow, Russia

Abstract. Finding an interpretable non-redundant representation of real-world data is one of the key problems in Machine Learning. Biological neural networks are known to solve this problem quite well in unsupervised manner, yet unsupervised artificial neural networks either struggle to do it or require fine-tuning for each task individually. We associate this with the fact that a biological brain learns in the context of the relationships between observations, while an artificial network does not. We also notice that, though a naive data augmentation technique can be very useful for supervised learning problems, autoencoders typically fail to generalize transformations from data augmentations. Thus, we believe that providing additional knowledge about relationships between data samples will improve model's capability of finding useful inner data representation. More formally, we consider a dataset not as a manifold, but as a category, where the examples are objects. Two these objects are connected by a morphism, if they actually represent different transformations of the same entity. Following this formalism, we propose a novel method of using data augmentations when training autoencoders. We train a Variational Autoencoder in such a way, that it makes transformation outcome predictable by auxiliary network in terms of the hidden representation. We believe that the classification accuracy of a linear classifier on the learned representation is a good metric to measure its interpretability. In our experiments, present approach outperforms β-VAE and is comparable with Gaussian-mixture VAE.

Introduction

The performance of machine learning algorithms crucially depends on the representation of data they work with. Relevant for the particular task data representation can significantly improve the performance of machine learning models on this task. As it was shown by Schmidhuber el al. in [1], Bengio et al. in [2] and I. Higgins et al. in [3], disentangled (or statistically independent) representations are particularly valuable, because they are useful in a wide range of applied tasks, particularly in computer vision field.

Advanced supervised machine learning models are known to be able to learn high-abstract relevant non-redundant representations, but they require specific preprocessing and a large amount of labeled data to achieve this. However, labeled data are usually limited and expensive to obtain. That is why the problem of learning good data representation from unlabeled data is important.

The requirements to amount of data required for learning good representation can be softened by using data augmentation technique. Initial dataset can be expanded by applying set of transformations to each training sample, where each of

these transformations is known a priori to not affect desired output. However, when using data augmentation, we still need decent initial amount of labeled data, and only labeled data can be used.

A frequent situation in practice is when large number of unlabeled samples is available, but labeling them is expensive. While data augmentation helps to significantly improve results in supervised tasks and learn good data representation during training, it usually does not help to learn more meaningful representation when applied naively in unsupervised tasks.

To see the problem, consider an example of CNN autoencoder for digit images, and assume we want it to extract an interpretable compact image representation, such that we can recognize digit from it easily (by linear classifier) regardless of rotation, scale, etc. If we provide our model an image of 3 and the same image, rotated 45 degrees, it is really hard to infer relation between these images from training signal (pixel-wise reconstruction loss). In a fact, in terms of pixel L_2 distance (typically used as reconstruction loss) 3 is closer to 9 than to 45-degrees rotated 3 (contrary to classification task, where rotation invariance is implicitly encoded in training signal). Because of this, applying rotation augmentations will help the autoencoder to memorize all rotated variations of digits, but it still fails to generalize image classes over rotations. This case was experimentally investigated in [4]. This kind of generalization we want to obtain can be equivalently formulated as disentanglement of class-related part of image description from rotation related part in terms of linear subspaces, i.e. factoring by rotation group.

More generally, when we work with natural images, we want to separate image description space into two subspaces of features: the feature subspace, representing inherent properties of the depicted objects themselves and the feature subspace, representing different kinds of transformations (zooms, rotations, shifts etc.).

One particular approach to such a separation was proposed in [4].

By using properties of the Group Equivariant Convolutional Networks, the autoencoder with specialized architecture can factor its internal representation space by group of transformations used by group equivariant convolutions in encoder network, allowing to distill all invariant information. It is shown that such representation can successfully disentangle image classes, making them linearly separable once irrelevant transformation groups are factored out. However, that approach has number of disadvantages, arising from reliance on group equivariant convolutions: group equivariant convolution operation needs to be implemented manually for desired group, which is not always possible or require additional tricks, set of transformations need to be proper algebraic group with known structure, computation and memory costs grow with group size.

In this paper, we introduce new, more flexible approach, inspired by observations on how biological brains do unsupervised learning. Instead of using specialized architecture to factor particular group of transformations, we change learning process to train encoder to produce more predictable image descriptions. To do so, we go away from training autoencoder on i.i.d. samples, and instead train it on conditionally connected samples, represented as tuples: image, transformation, result.

1. Biological Inspiration

Biological neural networks are known to learn good data representation in unsupervised manner. We suppose that one of the main reasons for it is that brain always learns in context. It remembers what it has seen before and tries to predict what will happen next. Particularly, a brain of an animal always knows where the animal is moving at every moment due to the signals it exchanges with the muscles.

Incredible importance of such a context was shown in paper [5].

They described a biological experiment, which was performed on pairs of newborn kittens at early stage of their post-natal development, which is considered as crucial for their learning. Each pair was placed into an apparatus, shown on Fig. 1. One of the kittens (active subject, A) had relative freedom to move actively around axes a, b and c. Another one (passive subject, P) was fixed inside of gondola, which repeated movements of A automatically, due to mechanical transmission system. Passive subject couldn't influence its own movements. In the same time, both subjects could equally observe their environment.
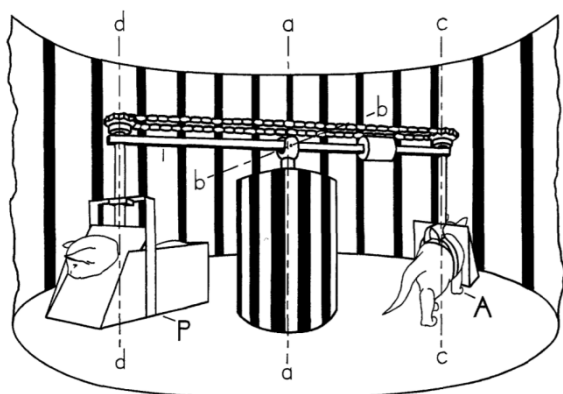
Fig. 1. Apparatus for equating motion and consequent visual feedback for an actively moving (A) and passively moved (P). Reprinted from (R. Held, A. Hein, 1963)

After spending suitable amount of time inside this apparatus, kittens were released, and their responses on visual stimuli were tested. Passive kittens didn't manage to react properly on approaching objects and obstacles, as if they didn't see them properly, despite the fact, that their eyes were healthy. In the same time, active kittens didn't have such a problem. They developed normal response on visual environment. Active kittens were able to learn relationship between their own movements and the visual data they received, and this context turned out to be crucial for them to learn, how to interpret visual data in a meaningful way.

But unsupervised machine learning models usually don't have such a context. Their sources of information are limited to distinct data samples from a training set via the loss function. It doesn't have context that biological brain has. Autoencoder models «know» that there is some compressed data representation and that they "should" find it, but found compressed representation isn't necessarily meaningful. Moreover, the task of compressing data can be performed fully without understanding the data structure. For example, JPEG algorithm is able to compress images, but is fully unable to extract useful features from it.

Having in mind the biological analogy, we think, that providing an additional context to autoencoder would let it learn more abstract and meaningful features. Particularly, we want to teach it the link between transformation (movement) and a new picture, which it gets after the movement.

2. Related Works

There are several different approaches to the described problem known so far. Two especially influential of them are β-VAE [5] and learning factorial codes by predictability minimization [3].

Among more recent approaches, we want to highlight Multi-Modal Deep Clustering [6], Maximizing Mutual Information Across Views [7] and Contrastive Learning [8].

Let us briefly highlight the insights of these works.

2.1. Factorial Codes

Factorial Codes are a particular variation of lossless encoding. A factorial code of a given data piece is a vector of statistically independent features, which can be decoded into that original piece of data again. In other words, that vector should get rid from statistical dependencies inside the data piece, but save all the information, which it contained, in the same time.

A particular way for unsupervised learning of such a non-redundant representation was introduced in [3]. There authors consider an abstract learner, which learnes a representation of the data, using some hidden units. Then they assign an adaptive predictor to each hidden (representation) unit of this learner. Each predictor learns to predict the output of the corresponding unit using the outputs of all other units of the same representational layer. In turn, each representational unit tries to minimize its own predictability. This competition encourages each representational unit to learn concepts, statistically independent of those upon which the other units focuse.

Authors also demonstrate particular implementation of this idea, performing experiments on particular neural network architecture and a learning algorithm for it.

2.2. β-VAE

Authors of β-VAE adapted well-known variational autoencoder [9] model for learning factorized interpretable data representations. To achieve this, they specify disentangled metric and use it to enhance VAE's loss function in a way to make it consider the «entanglement» as a part of the loss.

This enhanced loss function includes a hyperparameter β, which balances latent channel capacity

and independence constraints with reconstruction accuracy (and which gives the model its name). The edge case of $\beta = 1$ gives classical VAE model. The further β parameter is from 1, the more the "entanglement" part of the loss matters.

2.3. Multi-Modal Deep Clustering

In [6] a new algorithm for training a CNN-based neural network for natural images clustering is proposed. This algorithm combines main, unsupervised task of "noise as target" encoding with auxiliary task of rotation prediction in order to obtain a representation with meaningful clearly defined clusters.

"Noise as target" method aligns embeddings of images with target points sampled from a Gaussian Mixture. For this purpose, it utilizes loss function, based on distance between image embedding and closest target point. Additional task of rotation prediction encourages rotations of the same image to map into close vectors in the representation layer.

2.4. Maximizing Mutual Information Across Views

Authors of [7] encourage their model to maximize mutual information between features extracted from independently-augmented copies of the same image. The purpose of using such augmentations in this particular work is to create multiple views of a shared context, and then to force neural network to extract the most informative features, shared by all these views.

2.5. Simple Framework for Contrastive Learning

Authors of SimCLR [8] also aim to extract transformation-independent representations of visual data. They achieve this goal by combining original architecture, composed of encoder and auxiliary network, with suitable dataset augmentation. It's easy to see, that our approaches share much in common; however, they have deep differences as well. We will point out these differences below, right after giving brief outline of Contrastive Learning framework itself.

In SimCLR convolutional encoder takes two different transformations of the same image as independent inputs and learns representation of each one inside of its latent space. In turn, auxiliary fully-connected network, called "projection head", learns

to map (project) representations, corresponding to transformations of the same images, into vectors, close in terms of cosine similarity. Both networks train together by minimizing normalized temperature-scaled cross entropy loss through gradient descent (see original paper [8] for more details).

In our work auxiliary network plays another role: it learns transformations instead of projections. Besides, [8] operates with representations, which drop information about transformations, while our model aims to save information about transformations in a disentangled manner. It is achieved by basing our model on autoencoder instead of just encoder.

All these results have shown the way in which the resulting data representation is useful in a variety of tasks, especially in the computer vision field. However, first two papers attempt to approach the problem as solely a part of a compression task. Recognizing the huge importance of their results, we nevertheless admit that those techniques are not sufficient to learn good representation of more complicated datasets. In turn, the last three papers revolve around extraction of transformation invariants, but without keeping information about transformations themselves in a latent layer. So, despite the fact that such representations are proved to be useful, they are not decodable.

Our model, on the other hand, uses these invariants for disentangling learned representation, keeping it decodable. To achieve this, we augment the underlying VAE model in such a way, that it not only solves the compression task, but also tries to predict certain objects transformations. We see it as a part of important learning context, discussed above, which β-VAE and factorial codes models lack by themselves.

We also provide formal mathematical description of our idea, using Category Theory language, as well as Bayesian inference, justifying our architectural solutions.

3. Model Description

3.1. Categorical Background

Since we want to take into account the relationships between samples in our dataset, we suggest describing our dataset in the terms of the Category Theory. To begin, we recall the base definition of the category.
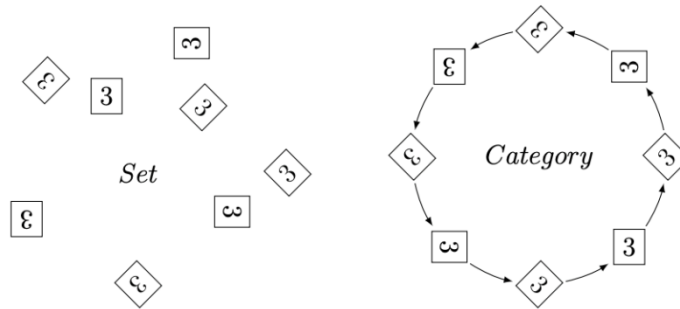
Fig. 2. Set versus Category

Definition. A category C is an aggregate of two classes - the class of objects $Ob(C)$ and the class of morphisms $Hom(C)$ with the following properties:

1. Each pair of objects $a, b \in Ob(C)$ corresponds to a set $H_{C(a,b)} \subset Hom(C)$ of morphisms.

2. Each morphism $\alpha \in Hom(C)$ belongs to one and only one set $H_C(x, y)$, where x and y are some objects from $Ob(C)$.

3. There is a composition operation defined in the class $Hom(C)$. The composition of $\alpha \in H_C(a, b)$ and $\beta \in H_C(b, c)$ gives a morphism $\beta\alpha \in H_C(a, c)$. This operation is associative, i.e. $\gamma(\beta\alpha) = (\gamma\beta)\alpha$ for any three morphisms $\alpha \in H_C(a, b)$, $\beta \in H_C(b, c)$, $\gamma \in H_C(c, d)$.

4. Each set $H_C(a, a)$ contains a special morphism 1_a, such that for any morphisms $\alpha \in H_C(x, a)$ and $\beta \in H_C(a, y)$ $1_a \alpha = \alpha$ and $\beta 1_a = \beta$. This special morphism is called the identity morphism of object a.

For more detailed explanation, see [10].

Returning to our situation, let's consider the set **X** of observations and the set **A** of allowed transformations between these observations. For example, **X** can be the set of some natural images, and **A** – the set of possible rotations, shifts and rescaling of each object. Now, we introduce a category X in a following way:

• we use the objects from **X** as objects of X: $Ob(X) = \mathbf{X}$;

• the pair of objects x_1 and x_2 from $Ob(X)$ is connected by a morphism $\alpha \in H_X(x_1, x_2)$, if and only if there exists a transformation $a \in \mathbf{A}$, such that $x_2 = a(x_1)$;

• the composition of morphisms in X is consistent with the composition of transformations from **A**.

Our main object of interest is, however, the latent space **Z** of our model, since we suppose that each object from **X** is generated by some hidden variable $z \in \mathbf{Z}$.

Let's introduce a category Z in a following way:

• we use the objects from **Z** as objects of Z;

• we say, that there is a morphism $a' : Z_1 \rightarrow Z_2$, if there is a morphism $a : X_1 \rightarrow X_2$. We denote this new set of morphisms $Hom(Z) = \mathbf{A}'$.

Let's also denote the embedding, learned by our model, by $P : \mathbf{X} \rightarrow \mathbf{Z}$.

Now, let's consider the diagram on Fig. 3. The goal of our model in the language of category theory is to make this diagram commutative. I.e. for each $x_1 \in \mathbf{X}, a \in \mathbf{A}$, we want the existence of $a' \in \mathbf{A}'$, such that the equality $a'(P(x_1)) = P(a(x_1))$ holds. To achieve this, we are using graph model, depicted on Fig. 4.
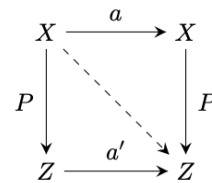

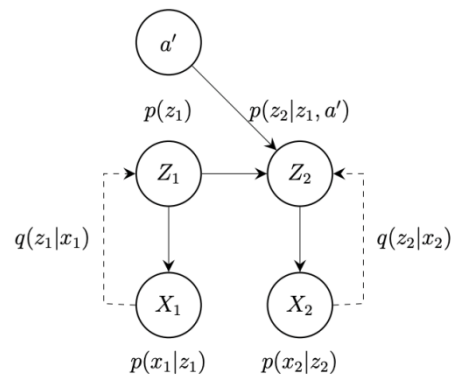Fig. 3. Diagram we want to be commutative

Fig. 4. Graph model

3.2. Variational Inference

Let us first remind the main steps of the variational inference for the classical Variational Autoencoder [9] and then derive it for our modified model.

Recall the identity (see, for instance, [11])

$$\mathbb{E}_{z \sim q(z|x)}[\log p(x|z)] - \mathcal{D}_{\text{KL}}[q(z|x)||p(z)] = L(x),$$

which holds for any distribution $q(z|x)$ over the same domain Z as $p(z|x)$. Here

$$L(x) = \log p(x) - \mathcal{D}_{\text{KL}}[q(z|x)||p(z|x)]$$

is the evidence lower bound (ELBO) and

$$\mathcal{D}_{\text{KL}}[p_1(z)||p_2(z)] = \mathbb{E}_{z \sim p_1(z)} \log \frac{p_1(z)}{p_2(z)}$$

is the Kullback–Leibler divergence between two distributions. Since the latter is always non-negative one can obtain the following lower bound of the evidence from the definition (2):

$$\log p(x) \geq L(x).$$

Moreover, according to (1), in order to maximize the data likelihood (with respect to some trainable parameters θ hidden in $p(x|z)$ and $q(z|x)$ one has to maximize objective function $L(x)$ along with minimizing the KL-divergence between $q(z|x)$ and $p(z|x)$. As usual, we assume that our encoding model $q(z|x)$ is complex enough to nullify the divergence term without affecting the log-likelihood term, so it suffices just to maximize $L(x)$.

In our case the space **X** of all observable variables consists of triples $(x_1, x_2, a) \in \mathbf{X} \times \mathbf{X} \times \mathbf{A}$ of two images and a transformation. We want our generative process to produce triples for which $x_2 = a(x_1)$ more likely than other, thus we want to maximize $p(x_1, x_2, a)$ for these points. The space Z of all latent variables consists of ples $(z_1, z_2, a') \in \mathbf{Z} \times \mathbf{Z} \times \mathbf{A}$. Here a' is a formal duplicate of observable variable a in the latent space involved in the generative process. Recall that the objective of auxiliary encoding function $q(z|x)$ is to give us distribution over z values that are likely to produce x, but its form is completely a matter of our choice. We use the following one:

$$q(z_1, z_2, a'|x_1, x_2, a) = q(z_1|x_1) q(z_2|x_2) p(a'|a).$$

The conditional distribution $p(a'|a)$ can be written in its explicit form

$$p(a'|a) = p(a|a') = \delta(a - a').$$

The equality that follows from our graph model depicted on figure 4 will also be useful:

$$p(x_1, x_2, a|z_1, z_2, a') = p(x_1|z_1) p(x_2|z_2) p(a|a').$$

Thus the whole generating process in our model is the following:

$$p(x_1, x_2, a) =$$
$$= \mathbb{E}_{z_1, z_2, a' \sim p(z_1, z_2, a')} p(x_1, x_2, a|z_1, z_2, a') =$$
$$= \mathbb{E}_{\{z_1 \sim p(z_1)\}} \mathbb{E}_{z_2 \sim p(z_2|z_1, a)} p(x_1|z_1) p(x_2|z_2) p(a).$$

In order to obtain the computable form of the evidence lower bound to train the model one has to substitute (5) and (6) into the identity (1) (written for specified above observable and hidden variables):

$$L(x_1, x_2, a) = \mathbb{E}_{z_1, z_2, a'}[\log p(x_1, x_2, a|z_1, z_2, a')] -$$
$$- \mathcal{D}_{\text{KL}}[q(z_1, z_2, a'|x_1, x_2, a)||p(z_1, z_2, a')] =$$
$$= \mathbb{E}_{z_1}[\log p(x_1|z_1)] + \mathbb{E}_{z_2}[\log p(x_2|z_2)] +$$
$$+ \log p(a) - \mathcal{D}_{\text{KL}}[q(z_1|x_1)||p(z_1)] -$$
$$- \mathbb{E}_{z_1} \mathcal{D}_{\text{KL}}[q(z_2|x_2)||p(z_2|z_1, a)].$$

Here expectation $\mathbb{E}_{z_1, z_2, a'}$ is taken over distribution $q(z_1, z_2, a'|x_1, x_2, a)$ and expectations $\mathbb{E}_{z_i}$ over distributions $q(z_i|x_i)$ respectively.

In our experiments we did not use naturally augmented dataset $\overline{\mathcal{D}} = \{(x_1^i, x_2^i, a^i) \mid i \in I\}$, but did the augmentation by picking a transformation a from finite predefined set $A = \{a^i \mid i \in I'\}$. Thus, given sampled $z_i \sim q(z_i|x_i)$ and $a \sim p(a) = \mathcal{U}_A(a)$ for every data point $x \in \mathcal{D}$, the objective (8) to maximize takes the form

$$L(x, a(x), a|z_1, z_2) =$$
$$= \log p(x|z_1) + \log p(a(x)|z_2) + \log p(a) -$$
$$- \mathcal{D}_{\text{KL}}[q(z_1|x)||p(z_1)] -$$
$$- \mathcal{D}_{\text{KL}}[q(z_2|a(x))||p(z_2|z_1, a)].$$

Here constant (with respect to trainable parameters) term $\log p(a)$ does not affect the maximization and can be omitted. However, one has to keep it if the distribution over possible transformations $p(a)$ is trainable itself.

3.3. Details of Implementation

More specifically, our model consists of two CNN encoders and two CNN decoders with the common parameter set and one full-connected network, which learns transformations (i.e. morphisms a' from A') in their latent space **Z**. All these networks are learning together via gradient descent

with Adam optimizer, maximizing variational lower estimate of data likelihood (9).

In our particular implementation we used the following set A of allowed transformations:
- up, down, left and right shifts by 3 px and 6px;
- scaling with coefficients 1.15 and 1.32;
- rotations by $12°$ and $24°$.

We assigned unique number for each transformation from this list. After it, at each training step we pick an object x_1 from our dataset with a particular transformation a from **A** and obtain ject x_2 by applying a to x_1. Next, we compute the output of variational autoencoder on both objects – x_1 and x_2.

Simultaneously we estimate a probability $p(z_2|z_1, a)$ of the object z_2 given z_1 and its desirable transformation a with fully-connected network, mentioned above.

4. Experiment Results

In this section we compare the proposed model with several standard unsupervised algorithms that allow one to obtain a compressed representation of data in the latent space. We use the achievable accuracy of a linear classifier in the latent space as a measure of disentanglement of the obtained representation. We perform all the experiments on the MNIST dataset.

4.1. 2-Dimensional Latent Space

To demonstrate the distinctive features of the used algorithms we display the MNIST dataset embedded in 2-dimensional latent space by each of them (Fig. 5 – Fig. 8). To explore stability of such representation we also encode a sample of each of the ten handwritten digits from the dataset rotated 360° with the step 3°. We call the obtained trajectories the orbits of these digits in the latent space.
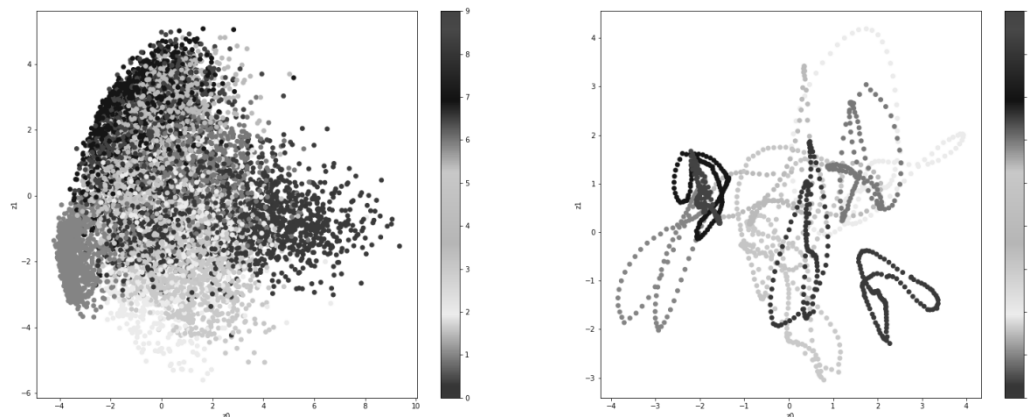

Fig. 5. PCA embedding (first two components)
on the left – the MNIST dataset in the latent space, on the right – an orbit of each sample digit in the latent space
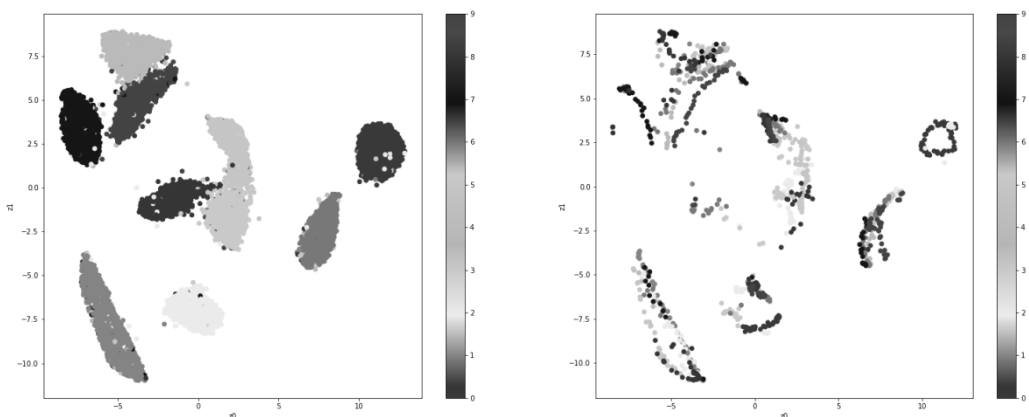

Fig. 6. UMAP embedding (with default hyperparameters)
on the left – the MNIST dataset in the latent space, on the right – an orbit of each sample digit in the latent space
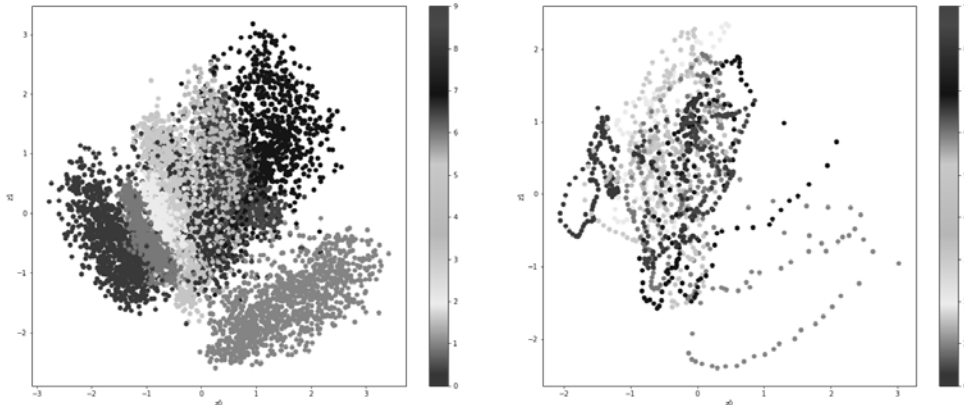
Fig. 7. VAE embedding

on the left – the MNIST dataset in the latent space, on the right – an orbit of each sample digit in the latent space
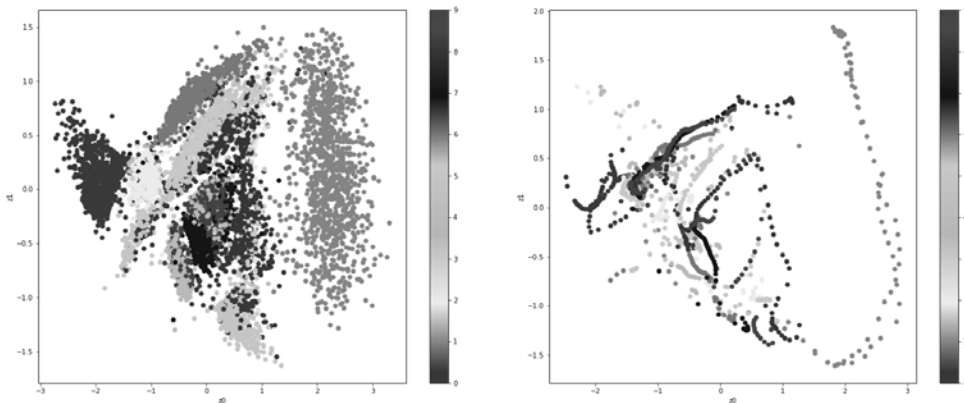

Fig. 8. Our embedding (with hyperparameters $\beta = 0.01, \gamma = 5.0$)

on the left – the MNIST dataset in the latent space, on the right – an orbit of each sample digit in the latent space

Table 1. Accuracy of a linear classifier on the 2-dimensional latent space of different algorithms

Algorithm	Linear separability
PCA	44.7%
UMAP	95.5%
VAE	63.0%
Our	65.7%

One can see measured accuracy on the testing set of a linear classifier trained on the training set embedded in the latent representation by each algorithm in the Table 1. Our algorithm along with classical VAE is not well-suited for data visualization task, thus it is outperformed by UMAP [12], which is.

4.2. 64-Dimensional Latent Space

Once the latent space is large enough, our algorithm outperforms all the baselines and allows us to obtain better visualization (by projecting onto the first two principal components in the latent space). The smoothness and stability of the orbits indicates that the latent code changes well-predictable under geometric transformations.

Conclusion and Future Work

In this paper we have described a particular approach that allows a model to learn transformation-aware representation of real-world objects without using labeled data. We have also shown that our model outperforms other well-known models in high-dimensional latent space in some aspects. We have provided mathematical generalization of the proposed principle, using the language of the Category theory, and described Bayesian inference for our model.

We believe that our ideas can be developed further in several directions and can be useful in other applications:

• Being generalized to the wider class of transformations (beyond Gaussian), our model can learn to perform stochastic multi-modal transformations properly. Now if new objects appear on the scene after applying particular transformation, our model tends to predict many inconsistent intermediate states, that look like partially appeared objects. One of the possibilities here is to adapt Generative Adversarial Network for our purposes, or use variations of VAE-like models with more flexible distribution families.

• Learning geometry of observed world and unsupervised scene understanding. Trained on a video sequence labeled with observer's movements, our model will require deep understanding of the scene in order to predict the next frame of the video. Such labeled observation sequences may be automatically generated and extracted from 3D simulations. In this class of tasks multi-modal probability models described above can be especially useful.
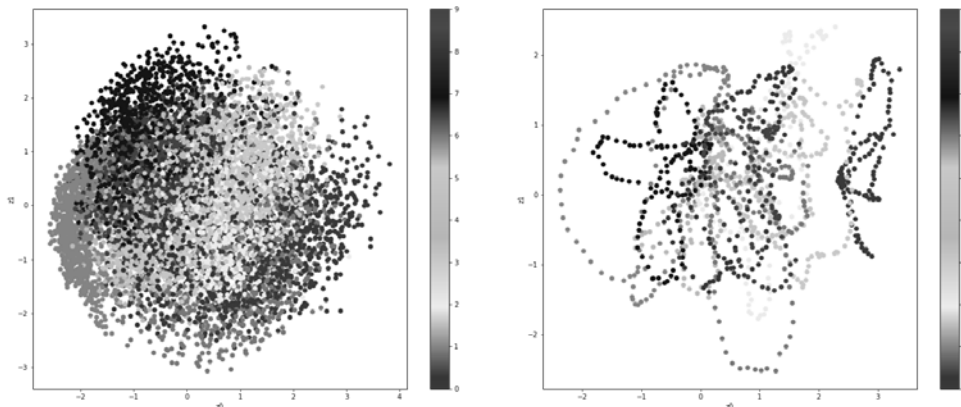

Fig. 9. First two principal components projection of the 64-dimensional VAE embedding
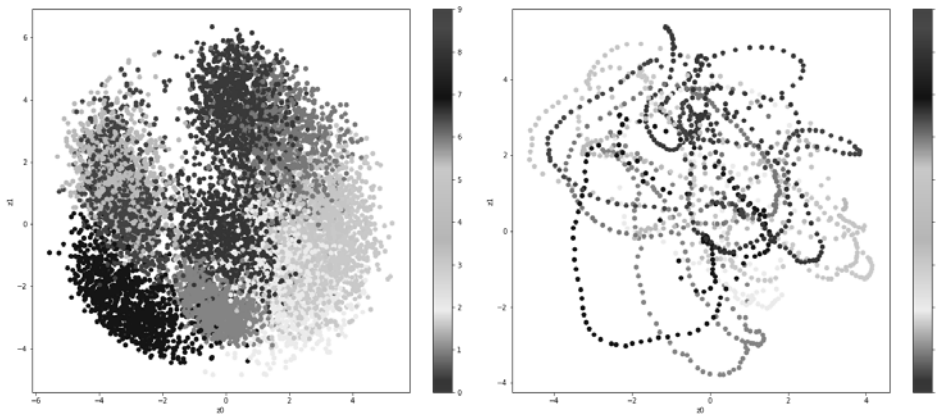on the left – the MNIST dataset, on the right – an orbit of each sample digit

Fig. 10. First two principal components projection of our 64-dimnesional embedding
on the left - the MNIST dataset, on the right - an orbit of each sample digit

Table 2. Accuracy of a linear classifier on the 64-dimensional latent space of different algorithms

Algorithm	Linear separability
PCA	91.7%
UMAP	96.0%
VAE	96.2%
Our	98.4%

References

1. Schmidhuber, J. H., «Learning factorial codes by predictability minimization», Neural Computation, т. 4(6), № 863-879.

2. Bengio, A. Courville, and P. Vincent, «Representation learning: A review and new perspectives», IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013.

3. I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, «β-VAE: Learning basic visual concepts with a constrained variational framework», ICLR, 2017.

4. Denis Kuzminykh, Daniil Polykovskiy, Alexander Zhebrak, «Extracting Invariant Features From Images Using An Equivariant Autoencoder», Proceedings of The 10th Asian Conference on Machine Learning, т. 95, № 438-453, 2018.

5. R. Held, A. Hein, "Movement-produced stimulation in the development of visually guided behavior," Journal of Comparative and Physiological Psychology, vol. 56(5), no. 872-876, 1963.

6. Guy Shiran, Daphna Weinshall, «Multi-Modal Deep Clustering: Unsupervised Partitioning of Images», arXiv:1912.02678, 2019.

7. Philip Bachman, R Devon Hjelm, William Buchwalter, «Learning Representations by Maximizing Mutual Information Across Views», arXiv:1906.00910, 2019.

8. Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton, «A Simple Framework for Contrastive Learning of Visual Representations», arXiv:2002.05709v1, 2020.

9. Diederik P. Kingma, Max Welling, «Auto-Encoding Variational Bayes», International Conference on Learning Representations, 2014.

10. Mac Lane, Saunders, «Categories for the Working Mathematician», Graduate Texts in Mathematics. 5 (Second ed.). Springer, 1998.

11. Doersch, C., «Tutorial on Variational Autoencoders», arXiv:1606.05908v2, 2016.

12. Leland McInnes, John Healy and James Melville, «UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction», arXiv:1802.03426v2, 2018.

Kuzminykh D. V. BS, assistant professor. Published works: 3. Scientific interests: Bayesian methods, Artificial Neural Networks, Meta-Learning, Game Theory. Moscow Institute of Physics and Technology (MIPT), 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russian Federation. Email: kuzminykh@phystech.edu

Kushnareva L. P. MS. Published works: 2. Scientific interests: Topological Data Analysis, Algorithms, Interpretability in Machine Learning. MS. Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russian Federation. Email: kushnareva.laida@gmail.com

Grigoryev T. A. MS. Published works: 3. Scientific interests: Bayesian Neural Networks, Time Series, Lie Algebras, Quantum Groups, Knots Invariants. Moscow Institute of Physics and Technology (MIPT), 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russian Federation. Email: grigorev.ta@phystech.edu

Zatolokin A. V. Assistant professor. Published works (since 2019): 1. Scientific interests: Artificial Neural Networks, Machine Learning, Learning Methods in ML. Moscow Institute of Physics and Technology (MIPT), 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russian Federation. Email: zatolokin.av@mipt.ru