Unsupervised Generative Learning with Handwritten Digits

Serge Dolgikh (serged.7@gmail.com)
National Aviation University

Research Article

Keywords: Machine learning, unsupervised learning, representation learning, concept learning, clustering

Posted Date: April 12th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1549651/v1

Unsupervised Generative Learning with Handwritten Digits

Serge Dolgikh [0000-0001-5929-8954]

Dept. of Information Technology,
National Aviation University, Kyiv, Ukraine
sdolgikh@nau.edu.ua

Abstract. Representations play an important role in learning of artificial and biological systems that can be attributed to identification of characteristic patterns in the sensory data. In this work we attempted to approach the question of the origin of general concepts from the perspective of purely unsupervised learning that does not use prior knowledge of concepts to acquire the ability to recognize common patterns in a learning process resembling learning of biological systems in the natural environment. Generative models trained in an unsupervised process with minimization of generative error with a dataset of images of handwritten digits produced structured sparse latent representations that were shown to be correlated with characteristic patterns such as types of digits. Based on the identified density structure, a proposed method of iterative empirical learning produced confident recognition of most types of digits over a small number of learning iterations with minimal learning data. The results demonstrated the possibility of successful incorporation of unsupervised structure in informative representations of generative models for successful empirical learning and conceptual modeling of the sensory environments.

Keywords: Machine learning, unsupervised learning, representation learning, concept learning, clustering.

1 Introduction

Representation learning has a well-established record in the discipline of machine learning. Informative representations obtained with Restricted Boltzmann Machines (RBM) and Deep Belief Networks (DBN) [1, 2], different flavors of autoencoders [3,4] and other models of unsupervised learning (unsupervised feature extraction) allowed to improve accuracy of subsequent supervised learning with conventional methods [5].

In experimental studies, a number of interesting results were reported, including the "cat experiment" that demonstrated spontaneous emergence of concept sensitivity on a single neuron level in unsupervised deep learning with images [6]. Disentangled representations were produced and studied with generative models of a deep variational autoencoder architecture and different types of visual data [7] pointing at the possibility of a general nature of the effect. Geometric and topological structure of conceptual representations of images of basic geometric shapes strongly associated with

characteristic patterns in the training data were produced and studied in [8]. In a growing number of results, concept-associated structure has been observed with real-world data of different types and origin, including Internet traffic [9], medical imaging [10] and other types and applications [11,12].

These results suggest that structure that emerges in the latent representations created by generative models in the process of unsupervised self-learning with minimization of generative error can be used as a foundation for learning methods and behaviors based on distillation of characteristic patterns in the observable environment, or "concepts", in an entirely unsupervised process, based on the ability to compress and restore the observations to and from informative compressed representations. Interestingly, these observations in unsupervised learning of artificial systems were paralleled very recently by a growing number of results in biologic sensory networks that demonstrated commonality of low-dimensional representations in processing of sensory information by mammals, including humans [13,14].

In this work we attempted to investigate the hypothesis that concepts can emerge in generative learning systems naturally under two constraints essential for any practical learning system in a complex sensory environment. The first one is the accuracy of representation, or modeling of the sensory information obtained from the environment. It provides a necessary foundation for intelligent behaviors that must be correlated with the environment to maximize the survival benefit. And the other one is the need to compress sensory information into a compact form so that it can be preserved to support intelligent behaviors in the future. A balance and dynamics of these factors, as we attempted to demonstrate in this work, can lead to emergence of a natural foundation for emergence of conceptual intelligence.

2 Materials and Methods

Generative neural network models in this work were based on the architecture of a convolutional autoencoder [15] with strong dimensionality reduction to a low-dimensional latent representation.

A particular architecture of generative models in the study was chosen based on several essential considerations. The first one being the universal approximation capacity of neural networks that makes them suitable for complex types of sensory inputs, including visual data. Secondly, the models were of minimal complexity in the sense of the size and use of specific and specialized architectural features. This consideration is essential from the perspective of generality of the obtained results. And finally, models of a similar type were shown earlier as successful in producing informative latent representations, including of image data [7-9,16].

General architectural structure of these models can be seen as "generic": they contain several standard components or blocks, serving different functions. The first one, in the ingress of the data to the model is physical rendering, serves as an adaptation of the physical sensory input to an invariant representation. In the models it was represented by a sequence of convolution / pooling layers to acquire higher scale features in the input images.

The second stage, depth, contained a single interconnected layer of a dimension D_d (in the study, $D_d = 50 .. 100$). Finally, the encoding or latent block contained a single sparse layer of a dimension $D_l = 20 – 24$, with activation sparsity penalty imposed in training. A combination of parameters of the rendering (R), depth (D) and latent (L) components thus fully describes a generic generative architecture, (R,D,L).

Whereas models used in this work can be considered as minimal from the perspective of generic architecture, more complex architectures are in no way limited to a fixed number or size, etc., of the layers and other features.

2.1 Deep Convolutional Autoencoder Model

A convolutional autoencoder architecture with a rendering stage of convolution-pooling layers followed by a single depth layer and a sparse latent layer of size 20 to 24 neurons (i.e., in the R,D,L notation, C2–3, 100, 20–24) was used to produce sparse latent representations of images in the training set, defined by activations of the neurons in the encoding (latent) layer. A sparsity penalty imposed in generative unsupervised training was essential to produce low dimensional representations of images, in most instances with significant activations of 2 – 4 latent neurons.

The decoding (generative) stage was fully symmetrical to the encoder. Overall, the model had 21 layers and approximately 86,000 trainable parameters. An architectural diagram of the model is given in Figure 1.

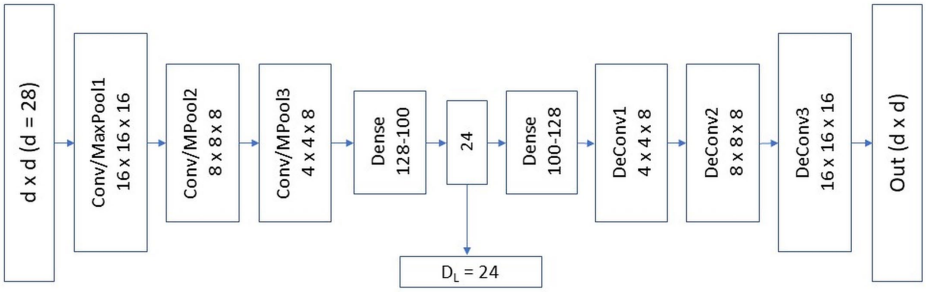


Fig. 1. Convolutional autoencoder with dimensionality reduction.

Architectural parameters of the models used in the study are described below.

Table 1. Model parameters

Rendering	Depth	Latent	Total layers	Trainable parameters	Cost function
Convolution 2-3 stages	1, 100	1, 24	11-15	$\sim 9 \times 10^4$	MSE, CCE

The models were implemented in Keras / Tensorflow [17] and trained for minimization of the deviation between the training batches of images and their generations by the model (further referred to as generative error).

2.2 Data

A dataset of images of handwritten digits (MNIST, [18]) was used as a model of visual sensory inputs with learning models described in the preceding section.

The images are grayscale, 28 × 28 pixels and contain real handwritten digits produced by multiple individuals.

The dataset has three parts: training, verification and test. In this study, generative models were trained with a subset of the training set of 50,000 images; verification subset was used to produce the unsupervised landscape and iterative concept learning as described in the following sections. Finally, the performance of the concept classifiers was tested with the testing and training sets that were not used in the learning process of concept classifiers.

2.3 Training

A success of generative learning was verified by the change in the validation value of the cost function over the process of unsupervised training and the ability of trained models to regenerate a random subset of images of the types represented in the training dataset (Fig.2). A majority of approximately 75% of models in the training process were successful by both criteria.

It was found that the success of learning was determined by an early cost threshold at 3 – 5 epochs of training, with models achieving it generally succeeding in subsequent training though training time to a plateau varied between models; while those that did not achieve the threshold were mostly unsuccessful and unable to demonstrate a good reproduction of input samples.

In the training process, a sparsity activation penalty (L1 sparsity constraint) was imposed on activations in the latent layer to produce sparse representations. Sparsity allows to produce more structured, "granular" representations while reducing the effective dimensionality of representations, an essential objective of practical learning as discussed earlier.

Fig. 2. Verification of generative ability of trained models (top: input; bottom: generation).

The generative training process produced models capable of two essential transformations performed by encoding and generative submodels of a trained model. The encoding transformation E realized by the encoding phase of the model transforms a sample in the observable space O into the latent representation R. The generative transformation G operates in the opposite direction, from the latent representation to the observable space and is realized by the generating part of the model:

$$y = T_e(X) = E(X); \quad X' = T_g(y) = G(y) \tag{1}$$

where T_e is the tensor of the encoding submodel (Input, Latent), T_g: the tensor of the generative submodel (Latent, Out), Fig.1. The objective of generative self-learning is therefore to adjust the parameters i.e., weights and biases of the encoding and generative tensors T_e, T_g in such a way that for a representative observable sample X, the mean deviation of the observable generation $G(E(X))$ from the observable sample X is minimized in some metrics imposed in the observable space.

It is worth noting that as a result of unsupervised training, encoding and generative functions in (1) are, in fact, decoupled, so that not only an encoded image of a "real" observation X, $E(X)$ but any position y in the latent space R can be associated with an observable image via generative transformation $G(y)$.

2.4 Generative Latent Landscape

Following the objective of the study, an approach was developed that allowed to investigate the structure in the latent space by purely unsupervised methods that do not require knowledge of the semantics, concept, class or any other prior information about the observable data. The process of producing such unsupervised structure (or "generative landscape" of the representation, as referred to in the rest of this work) is based on identification of a density structure, such as density clusters in a general sample of encoded sensory inputs with methods of unsupervised density clustering [19,20].

The method is based on several essential assumptions. The first one is generative accuracy achieved in the process of unsupervised training. The second is sparsity of resulting representations, that provides two essential benefits: a lower dimensionality of encoded inputs, and more detached, or decoupled structure of representations making it easier to detect and harness for learning. And finally, an assumption on the composition of the training set, containing sufficient populations of a constant number of characteristic types of inputs (i.e., representativity).

In the implementation of the method described below, it was assumed that sparse representations of images produced by trained generative models have significantly lower dimensionality than the size of the latent layer, that is the maximum number of latent neurons with significant activations on inputs in the training set. This assumption is supported by a number of recent experimental results neuroscience [13,14] demonstrating commonality of low-dimensional neural representations in processing sensory inputs by animals and humans.

Then, assuming the sparse dimensionality of three, the activations of latent neurons in a trained generative model of described architecture on samples in a general representative subset of images, X_G can be distributed between three-dimensional "slices" of the 24-dimensional latent space identified by the indices of activated neurons (i_1, i_2, i_3): $x \in X_G \rightarrow E(x) \in S_x(i_1, i_2, i_3)$ (Fig. 3).

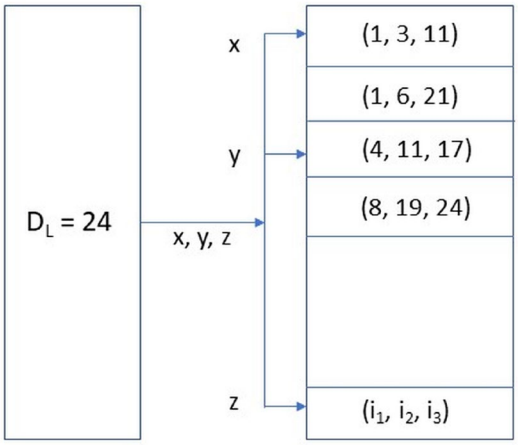


Fig.3. Stacked structure of latent activations

The algorithm of detecting an unsupervised conceptual latent structure (landscape) in the sparse representation space of a generative model is described in Table 2.

Table 2. Production of unsupervised density landscape

Step	Result	Process
1. Produce slice structure	A set of d=3 slices ordered by population	1.1. Produce encoded image of general sample, $E(X_G)$ 1.2. Allocate samples in $E(X_G)$ to 3D slices by highest activations: (S_k, E_k) [(1)]
2. Produce slice density landscape	A set of density clusters per slice	2.1. For a given slice, produce a projection of the slice sample E_k to slice coordinates. 2.2. Apply a density clustering method on the resulting 3-dimensional set of slice samples to obtain a set of density clusters, $D_k = \{ d_k \}$ 2.3. The set of density clusters D_k and trained density clustering method, M_k represent a slice landscape: $L_k(S_k) = (E_k, (D_k, M_k))$
3. Produce latent density landscape	A set of characteristic latent density structures	3.1. Repeat slice landscape definition for all slices found in the encoded general sample $E(X_G)$ [(2)]. 3.2. The resulting density structure, $L(X_G) = \{S_k, L_k = (D_k, M_k)\}$ is the unsupervised density landscape of the representation

[(1)] Satisfying the significant slice activation condition on the sum of slice activations: $\Sigma a_j \geq f$ $a_{max}, f = 0.3$ in the study.

(2) A truncated set of 32 slices (out of 2024 possible combinations in a 24-dimensional space) containing over 80% of the encoded general sample was used due to processing limitations.

With the density landscape *L(M)* produced in the described process by a trained generative model *M*, assuming that the general sample used in production is representative of the observable distribution, it is possible to associate an observed sensory input *y* to a density structure in *L(M)*, $d_k(y)$ as:

$$C_{nat}(y) = M_k\big(e_k(y)\big) = d_k(y) \tag{2}$$

where $e_k(y)$, a projection of the latent image of *y*, *E(y)* to the slice of the most significant activations. The latter can be interpreted as its natural, as opposed to externally defined, characteristic type, or concept of the observable sample.

In conclusion let us note some essential properties of the landscape method described in this section:

- It requires only a representative general sample of the input distribution and thus is completely unsupervised;
- Identification of the latent landscape can be performed with an encoded general sample, significantly reducing the required memory and processing capacity.

3 Results

3.1 Generative Structure of Latent Landscape

The first question that can be asked following generative training of the models and production of generative latent landscape is, is the landscape correlated with characteristic patterns in the observable data, represented by the training set? To address it, methods of generative probing and scanning developed in the earlier studies [8] were applied in the slices of the latent space identified with the landscape production method described above.

To "probe" a latent region represented by a set of latent points Y_l, the generative transformation *G* (1) is applied to the latent sample producing a set of observable images, $I_l = G(Y_l)$. An analysis of the resulting set, I_l can provide insights into the semantics of latent coordinates and distribution of characteristic patterns in the latent space. With generative scanning, probing is applied to a multi-dimensional grid in a latent region of interest.

With a generative landscape produced with one of the generative models, observable images were generated from the positions of the centers of landscape density clusters in the slices identified by the landscaping method, d_k (Fig. 4).

In the figure, slices in the latent dimensions are ordered by the population with significant activations in the slice, identified with a general representative sample, vertically top to bottom; density clusters are ordered in horizontal rows by population, left to right. It can be seen clearly that most density clusters identified by the method were associated with realistic images of digits, with all digits present in the top 20 slices. As well, certain "specialization" or distribution of different types of digits between different low-dimensional slices can be observed.

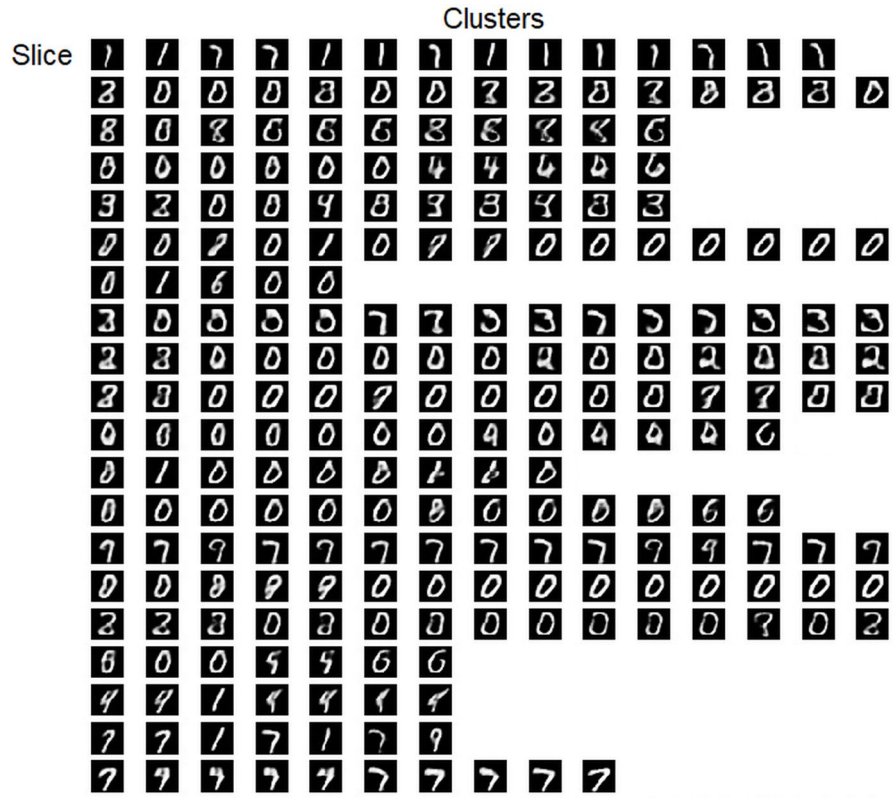


Fig.4. Generative structure of the latent landscape (20 slices, first 15 clusters)

The structure of the latent landscape described above provides a natural indexing of structural latent features, such as density clusters with a two-dimensional index (*slice*, *cluster*). For example, clusters associated with digit "4" in Fig. 4 can be indexed as: (4, 7-10), (18,1-2) and so on. It needs to be noted that this structure is specific to each individual learning model and there's no reason to expect that the index will have the same semantic meaning or even be valid for another model. Consistency of latent landscapes between learning models of the same architecture is discussed in Section 3.3.

Further analysis of the landscape of 32 × 16 top slices, clusters produced by the same model showed that out of 216 non-empty clusters identified by the method, 207 or ~ 96% were associated with identifiable handwritten digit forms.

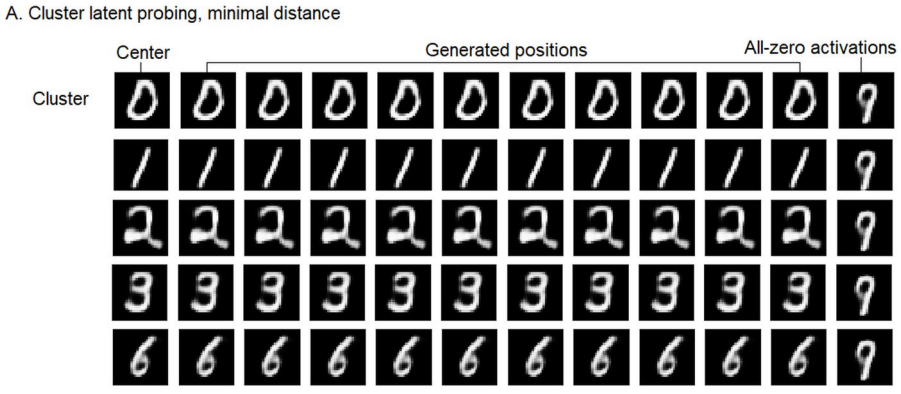
A conclusion from the results in this section is that latent landscape produced with representations created by generative models in the process of unsupervised self-learning can be associated with common patterns in the observed data.

3.2 Geometry and Topology of the Latent Landscape

The next question that can be asked relates to geometrical and topological structure of representations produced under the constraints of generative learning. Are characteristic latent regions connected? Is generative transformation continuous?

Generative probing again can provide an answer to these questions. In the next set of experiments, sets of random samples $Y_r(d_k)$ were produced on a three-dimensional sphere of a small radius r centered in the centers of the clusters of the landscape d_k, imitating small variations of latent positions along different latent axes. By producing observable images, $I_k(r) = G(Y_r(d_k))$ and comparing them with the generated image of the center position, observations can be made on the topology of latent regions and properties of generative transformation G.

Examples of resulting visualizations for different clusters at the same distance; and for the same cluster at different distances from the center are shown in Figure 5, A, B respectively (distances $d_1 - d_3$ relative to characteristic size of the latent distribution region measured as an average of the maximum latent axes values in a general representative sample; the first position in the sample corresponds to the center of the cluster, the last, all-zero activations).



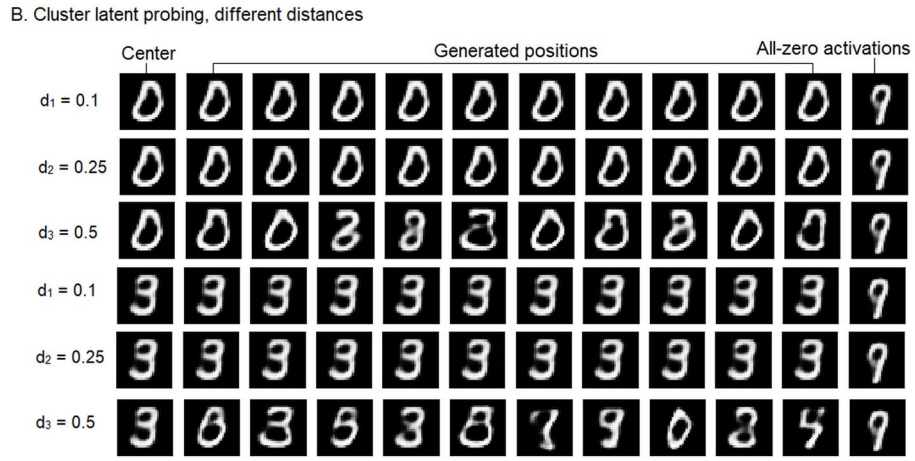


Fig. 5. Latent probing of landscape features. Distances: $d_1 = 0.05$; $d_2 = 0.1$; $d_3 = 0.15$ D_g

It can be observed that generated images of the "flow" of variations from the centers of identified clusters at different distances described a well-behaved continuous transformation of a latent position encoding specific type of an image to the observable space. This pattern was observed for all studied clusters, of both recognizable digits, and those of unrecognizable shapes.

The results of the experiments in this section suggest a well-defined geometry and topology of the generative mapping of a stacked latent space to observable images, with a structure of a density landscape that can be identified in an entirely unsupervised process, as described in Section 2.4.

3.3 Consistency of Latent Landscape

With a highly structured latent landscape described in the preceding sections, a question can be asked, how consistent are the characteristics of the resulting landscape between different, independently trained models of the same architecture and with the same or similar samples of sensory inputs?

To answer it, we performed an analysis of latent landscapes produced with three independently trained generative models, s24-1, s24-2, s24-3. The models were trained over 40-80 epochs with a training set of 10,000 samples, achieving a training plateau at validation loss of 0.12-0.14 (the starting value of 0.7) and good to excellent generative performance on a subset of images and were not selected by any specific criteria.

The measured characteristics were: the size of the landscape, i.e., the number of non-empty density clusters; recognition, the fraction of the landscape associated with recognizable digits, indicating a correlation of the landscape with the content of the training set; representation and content of the landscape, such as representation of all types of digits and distribution of types of digits between slices and clusters. The resulting measurements are presented in Table 3.

Table 3. Consistency of latent landscape between learning models

Model	Size	Recogni-tion	All digits represented	Highest population	Lowest population	Nil activation
s24-1	474	0.973	True	0,7,3	4,6,9	3
s24-2	396	0.975	True	0,7,1	2,5,6	9
s24-3	485	0.971	True	4,0,7	2,8,6	9

As can be seen in these results, produced landscapes were consistent in some characteristics, while having individual differences in the others. The size of the landscape is controlled by the bandwidth parameter of the clustering method, that was found to be in the same range (0.011-0.012) for all models, requiring only minor tuning (less than 5% of the average value). The recognition factor, representing the fraction of "real" recognizable digits in the landscape clusters was very close: average 0.973 with a standard deviation of 0.002. All model landscapes had a full representation of all types of digits, 0 to 9 although the observed distribution of the number of clusters to type of digit was far from uniform (Figure 6).

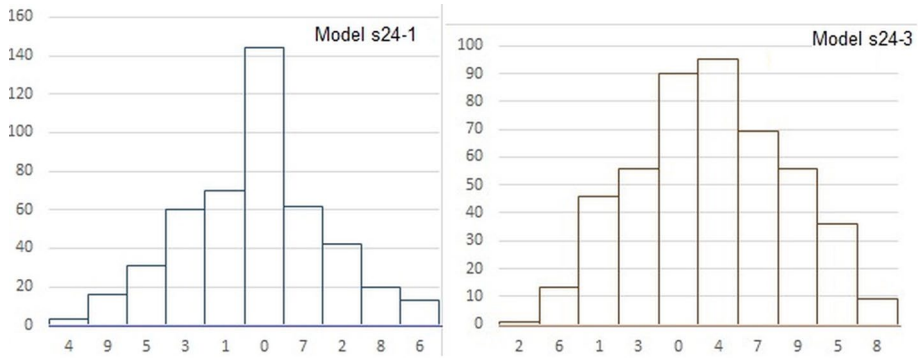


Fig. 6. Number of landscape clusters to digits, independently trained models

The individual differences were observed in the areas of distribution and encoding of digits in the landscape. While types of digits with the highest representation were mostly aligned, those of the lowest representation varied significantly (Table 3). The same observation applies to all-zero activation and encoding of digits to landscape, for example, "2" was encoded to clusters: (2; 14-15), (14; 11-14) versus (9; 9,12,15-16) by models s24-1 and s24-3 respectively. This confirms the comment made earlier that landscape indices are internal to the learning models and have no semantic meaning for other learners.

Overall, the results in this section demonstrated that latent landscapes in the representations of successful generative learners of similar architecture were consistent in some essential characteristics, allowing to use them to enable flexible and iterative learning from direct interaction with the sensory environment.

3.4 Learning with Latent Landscape

Based on the observations on the structure and consistency of the latent landscape that emerges in the process of unsupervised generative self-learning in the preceding sections, a question can be asked, whether it can be used to improve learning efficiency, in particular, achieve reasonable recognition of digits with minimal sets of known samples? Learning of this type, that is, iterative, empirical and driven by interactions with the environment is characteristic for natural, biological systems [21].

Let us consider an imaginary scenario where an attention of the learner was attracted, possibly as a result of some significant event, to a single instance of a novel pattern, not previously known, that is, it could not be identified as a known class or type from the available knowledge and/or previous experience. With a single positive instance of a class, conventional classification is not possible as there is no knowledge about "other", different classes; this is not necessarily the case with generative systems, as the latent landscape formed as a result of generative learning can also be considered an input to the learning process. A key assumption here is that either a) the new concept being learned has reasonable representation in the sensory environment and generative training set (i.e., "seen but not known" scenario) or b) has some mechanism of incorporating specific observations of interest or importance into the process of generative learning and production of the landscape.

The latter one is an interesting case that will be examined in a future study, and going forward it will be assumed that a concept already has certain representation in the landscape but has not yet been learned. In the case of sensory environments modeled by images of handwritten digits, this can be a scenario where some digits have been learned that is, can be classified with sufficient confidence, in both positive and negative samples to a known class, while others are not yet known.

As a result of the encounter, a generative learner under these assumptions would have: a) a single observable sample (y, r_y) of a new concept of interest C_y (observable and latent position); and b) the latent landscape $L(M)$ produced as a result of generative learning with a representative set of sensory inputs.

One can realize then that the structure in $L(M)$ allows to produce the first, initial iteration a classifier of C_y by using two types of latent samples:

- Positive (Y_p): samples associated with r_y via latent structure, for example, a set of samples in the density cluster associated with the latent position of the observation, $d_k(r_y)$.
- Negative (Y_n): clusters other than $d_k(r_y)$ can be considered, in the first learning iteration as those representing negative background with a selection of representative samples drawn from them.

Then, a binary labeled set associated with C_y can be produced as a combined set of sample-label pairs, $T_y = (Y_p, \text{True}) \cup (Y_n, \text{False})$ and a binary concept classifier K_y obtained by training one of the known classifiers with the set T_y produced in this process. Importantly, K_y can produce predictions for all observable samples x, positive and negative as:

$$p(C_y, x) = K_y(E(x)) \quad (3)$$

where $p(C_y, x)$: the probability of an observable x sample being of the type C_y.

An ability to make positive predictions of sensory inputs for a novel concept from a single positive instance, based on the structure of unsupervised generative landscape can be used as a basis for iterative learning process based on a sequence of empirical trials.

3.4.1. Minimal sample learning

In the realization of the method in this section, we used one true positive instance of a new concept and a number of "artificial" ones generated as follows:

- c_r, the center of the cluster $d_k(y_r)$ associated with the latent position of the encoded true instance, y. The cluster and its center position are determined by the clustering method M_k in production of the landscape, Table 2 as: $d_k = M_k(y_r)$ where slice association is by highest activations and the condition of significant activations (Table 2). If a direct match was not found, the first slice matching 3 out 4 most significant activations selected.
- m_r, the middle position on the line between c_r and y_r. The distance $d_r = ½ \parallel y_r - c_r \parallel$ thus represents a characteristic scale of the method.
- A number of positions (n_p) generated within the distance of d_r from c_r. This process produces a set of $n_p + 3$ positive samples of the concept represented by the initial "signal" instance y.
- The negative, "out of class" set was represented by center positions of clusters $d_j \neq d_k$, up to a certain maximum, m_n to maintain a balanced set.

The process produces a set of ($n_p + 3$, m_n) labeled concept samples for training of a binary concept classifier. In the implementation of the method, $n_p = 2$, $m_n = min(2 \times n_p$, card(D_k) - 1) was used with a nearest neighbor classifier [22].

And essential condition for the initial, zero iteration of a new concept classifier is not an excellent accuracy but rather:

a) the ability to produce positive predictions, as only they can produce empirical feedback, meaning that it is not strongly biased to rejection.
b) a certain minimal level of accuracy, so that new true samples of the concept are being discovered in empirical iterations.
c) a certain minimal level of specificity, meaning that the classifier is able to distinguish true instances of the concept from the background i.e., it is not overly biased to acceptance.

While a detailed study and discussion of these criteria in specific learning scenarios will be provided elsewhere due to limitations of this work, in the experiments with learning models over 70% of the initial iteration classifiers with a single true concept sample satisfied the condition of minimal sensitivity and specificity of the produced classifier of (20%, 55%) respectively, with different types of digits and different individual learning models. The mean initial learning performance observed in these experiments was (40%, 85%).

3.4.2. Iterative learning with latent landscape

The ability of the initial classifier to make positive predictions is essential for the success of an iterative learning process based on acquisition of true concept samples, both positive and negative, via a sequence of predictions and empirical interactions with the environment.

In learning iterations, a concept classifier produced in the previous iteration makes predictions on sensory inputs, and a subset of positive samples is tested after prediction. It is assumed that a trial produces a small set of true samples, positive (resulting from true positive predictions) and negative (false positive). The iteration samples then used to extend the concept training set as described below with both genuine samples produced in the empirical test and generated ones based on their positions and associations in the latent landscape. The learning iteration is completed with producing a new iteration of the concept classifier, retrained with the extended set of concept samples.

A detailed definition of the iterative learning process based on unsupervised latent landscape is provided in Table 3.

Table 3. Iterative learning process based on latent landscape

Step	Result	Process
Learning iteration (j-1): concept training set T_{j-1}, concept classifier K_{j-1}		
Learning iteration j		
1. Produce positive concept predictions	Iteration prediction set, S_j	1.1. Produce a sufficiently large set of positive predictions with a previous iteration concept classifier, K_{j-1} 1.2. Obtain encoded sample set $E(S_j)$
2. Empirical test	Subset of true positive and negative samples	2.1. By testing predictions in S_k, produce subsets of samples of a size n_I (iteration sample): true positive predictions, $Y_{p,j}$ and true negatives, $Y_{n,j}$.
3. Produce iteration training set of samples	Concept training set of the iteration, T_j	3.1. For each positive sample in $Y_{p,j}$: generate additional positive samples; and negative samples based on association to the landscape L(M) as with the initial learning process (positive label). 3.2. Append negative samples $Y_{n,j}$ (negative label). 3.3. Append the produced set to the training set of the previous iteration to obtain T_j.
4. Produce iteration concept classifier	Concept classifier of the iteration, K_j	4.1. Retrain concept classifier with the iteration training set, $K_j = K(T_j)$.

Importantly, the concept sample set in this method can be stored in the latent coordinates, with a massive reduction of required memory resources (for example, compression of 1,000 × 600 grayscale images to a three-dimensional sparse representation

would produce a reduction of required memory capacity of up to five orders of magnitude).

As a result of the iterative learning process, two objectives were achieved at the same time: a set of true concept samples, both positive and negative was acquired in empirical iterations that can be retained in the memory for example, for another learning process in the future; secondly, iterative extensions of the concept training set allow modeling of the concept distribution in the latent space with higher precision and consequently, improved accuracy of the concept classifier.

3.4.3. Iterative learning: evaluation of performance

In the implementation of the method in this work, $n_J = 5$ was used (i.e., 5 true positive and same number of negative samples of the concept per learning iteration), with 5 to 10 learning iterations, including initial, with the total of 51 to 91 true concept samples over the learning process.

The results presented below are indicative of the performance of the method, with a more detailed statistical analysis expected in another work. Two sets of measurements are presented:

1. Evaluation of the learning performance of a single generative model (i.e., same latent landscape), with an independent test set not used in the learning process, with all types of digits, over 10 independent sequences of 10 learning iterations, 91 true concept samples (Table 4.1).
2. Learning performance of different models with different types of digits, over 50 independent sequences of 10 learning iterations, 91 true concept samples (Table 4.2).

In the evaluation of the learning performance of models trained in the landscape-based process described in the preceding sections two types of classification decisions were used:

a) Categorical: True on a positive sample, False on a negative is considered a correct prediction and vice versa. Learning metrics: *cat.pos*: the rate of correct predictions on positive samples (true positives); *cat.neg*: the rate of correct predictions on negative samples (true negatives).

b) Probability-based: a positive probability of above 0.3 on a true sample and negative, above 0.71 considered a correct prediction. Metrics: *prob.pos*: the rate of correct predictions on classified samples; *prob.neg*: the rate of correct predictions on classified samples; *prob.conf*: the fraction of not classified samples.

Probability ranges were chosen based on the observation that positive predictions tended to be spread over the range while the negative ones, close to categorical. The positive and negative ranges were mutually exclusive, so that a sample could not be classified into both categories; however, that introduced a possibility of it not being classified into either category, i.e., a confusing observation, that was measured as well (parameter *prob.conf*). The choice of decision factors, again, was only indicative, and these parameters can be adjusted by learning systems to maximize learning performance.

Table 4.1. Landscape learning performance, model s24-2, 10 learning sequences

Digit, metric	cat.pos	cat.neg	prob.pos	prob.neg	prob.conf	prob.pos, stdev	prob.neg, stdev
"1"	0.973	0.972	0.984	0.961	0.003	0.001	0.004
"0"	0.702	0.952	0.963	0.887	0.006	0.003	0.004
"6"	0.722	0.965	0.882	0.908	0.024	0.038	0.023
"8"	0.454	0.967	0.908	0.727	0.013	0.004	0.006
"3"	0.542	0.866	0.734	0.888	0.021	0.033	0.009
"9"	0.422	0.971	0.967	0.665	0.012	0.024	0.072
Mean, all	0.642	0.935	0.907	0.827	0.014	0.014	0.017

For other digits, the learning metrics were in the range between the best and the lowest in the table.

Table 4.2. Landscape learning performance, different models, 50 learning sequences

Metric	cat.pos	cat.neg	prob.pos	prob.neg	prob.conf	prob.pos / prob.neg stdev
s24, all models	0.740	0.927	0.870	0.839	0.01	0.008 / 0.007

As can be seen from the results above, probability-based strategy allowed to improve positive accuracy at the cost of a smaller reduction in specificity. In practical artificial or biological systems these parameters can be attuned to maximize learning performance.

Overall, it can be concluded from the results presented in this section that landscape-based learning methods can produce a noticeable success in iterative empirical learning with minimal sets of samples. Optimization of the method was not an objective of this work and further improvement in learning performance can be achieved by tuning and optimization of the parameters.

4 Discussion

A possibility to harness structured generative representations of sensory data, including of more complex types such as real-world images with significant variation of content for successful environment-driven learning of concepts demonstrated in this work can be noteworthy from several perspectives.

First, it offers a direction for studying and modeling natural learning, methods and strategies based on direct observation and interpretation of the sensory environments; with minimal confident samples obtained in interaction with the environment; in a flexible process based on empirical trials that is not dependent on availability of known concept data upfront, before the learning process can begin. It can be hoped that models and systems designed on these principles can be more effective in the environments

where confident prior knowledge of the domain is not available, as well as contribute to investigation of evolution of intelligent functions and behaviors.

It can provide essential insights into the origins of higher-level concepts and conceptual intelligence. According to the results presented in this and a number of other works [6-9], concept prototypes can emerge in generative processing of sensory data as native structures in generative representations related to and correlated with characteristic common patterns in the sensory inputs. Essential conditions for emergence of such structured representations appear to be generative accuracy, that is, encoding sufficient information about the observed distributions in the latent space, and redundancy reduction [7].

The line of investigation based on unsupervised structure emergent in representations of successful generative models can point to a solution to the conceptual "chicken and egg" puzzle: if true instances of higher-level concepts are needed to analyze and determine their representations, how can they be defined and what is their origin? Methods of analysis of unsupervised generative representations discussed here allow to associate origins of general higher-level concepts in the sensory data with characteristic latent structures that can be determined with entirely unsupervised methods and without any prior knowledge of the concepts.

As the results in Sections 3.1, 3.2 suggest, concepts in this process may not emerge as a single broad class with subsequent specialization (hierarchical stratification). Rather, concept-associated features such as density clusters can be spread across the components of a complex latent space, such as stacked low-dimensional slices studied in this work. Some concepts can be associated with relatively small number of latent structures in the same low-dimensional slice (i.e., produced by a constant group of latent neurons). Other concepts can be distributed between different slices (Fig. 4), encoded by variable groups of neurons. Generalization of multiple prototypes such as concept-associated clusters into a single concept class can happen in a process of empirical learning as described in Section 3.4.

Let us consider a concrete example for an illustration of this point. Suppose we have a single positive instance of concept of interest, for example in the context of the work, an image a digit "2". There are different latent regions associated with different variation of handwritten representations of the digit by different individuals in the dataset. Further, suppose an early iteration of a classifier produced a positive prediction for a different version of the same digit, located in a different cluster, and / or slice. It is possible for example, due to relative proximity of the latent positions of the samples in the full latent space.

A positive prediction would cause an empirical test of the identified input sample. If the test confirms similarity of the outcomes (for example, similar amount of useful substance obtained in the trial), the sample can be recognized as another true positive representative of the concept, and the region of its distribution in the latent space can be updated to improve the accuracy of its description by the concept classifier. The process driven by empirical interaction with the environment can continue until confident recognition of the concept is achieved. In this process generality of concepts emerges as a synthesis of characteristic latent structure associated with common patterns in the

sensory data and empirical trials, from the lower, “flat” levels of latent structure up, rather than in the hierarchical model, top down.

Another observation points to the character of latent encoding and landscape learning as essentially geometrical in nature. While proximity-type classifiers such as kNN were capable of successfully leaning concept classes in an iterative process with minimal empirical samples (Table 4, Section 3.4.3) classifiers of several other types including neural network models such as perceptron [23] and SVM [24] were unable to interpret information encoded in the latent positions and produced strongly overfitted classifiers incapable of successful learning. Investigation of geometrical and topological properties of generative representations could provide further insights into learning processes based on generative latent structure.

Harnessing latent landscape in the initial learning phase when confident data can be very scarce allows to produce the initial, “signal” generation of concept classifiers with minimal, down to a single instance, learning sample. An essential condition for successful learning in this initial phase is certain minimal sensitivity, that is, the ability to distinguish suspected instances of the concept from the background. In the iterative learning process, true instances of, and outside of the concept type are acquired in empirical trials that can be expected to have certain cost for the learning system. As only positively predicted samples are tested, an initial sensitivity that is sufficient high allows to reduce the overall cost of learning and thus can be an essential selection criterion, while high accuracy and confident recognition of the novel concepts can have lower priority in this stage of the learning process.

For example, initial learning performance with accuracy and sensitivity observed with the generative models in this work (Section 3.4.1) would result in a substantial reduction of the cost of empirical tests, measured in empirical tests per true positive sample gained compared to the random selection strategy, the only viable alternative in a scenario with minimal known samples. The objective of a learning system in a realistic environment thus can be formulated as maximizing the accuracy of the concept resolution in both positive and negative channels, while minimizing the empirical cost of learning.

Generative models investigated in this work were of a generic type and limited complexity, in terms of equivalence of a nervous system on the level of simple organisms, like worms and jellyfish [25,26]. An ability of such minimal models to identify characteristic patterns in simple sensory environments may provide an argument for an earlier emergence of conceptual intelligence in biological systems. Consistent conceptual modeling of the sensory environment can form a basis for development of further intelligent functions and behaviors including collective intelligence [27].

Overall, the results of this work demonstrated that structured representations that emerge in the process of unsupervised generative learning with sensory inputs from the environment under the constraints of generative accuracy and compression of information can provide a natural platform for development of conceptual models of the sensory environments and intelligent behaviors based on such models. Given the range of models, architectures and data types where the effect of spontaneous categorization has been reported the conclusion that can be drawn from these results is that it

represents a natural general effect in the information processes in learning systems regardless of origin, biological or artificial.

Conflict of Interest

The authors have no conflicts of interest to declare.

References

1. Hinton, G., Osindero, S., Teh Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18(7), 1527–1554 (2006).
2. Fischer, A., Igel, C.: Training restricted Boltzmann machines: an introduction. Pattern Recognition 47, 25–39 (2014).
3. Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning 2(1), 1–127 (2009).
4. Welling M., Kingma D.P.: An introduction to variational autoencoders. Foundations and Trends in Machine Learning, 12(4), 307–392, 2019.
5. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of 14th International Conference on Artificial Intelligence and Statistics 15, 215–223 (2011).
6. Le, Q.V., Ransato, M. A., Monga, R. et al.: Building high level features using large scale unsupervised learning. arXiv 1112.6209 (2012).
7. Higgins, I., Matthey, L., Glorot, X., Pal, A. et al.: Early visual concept learning with unsupervised deep learning. arXiv 1606.05579 (2016).
8. Dolgikh, S.: Topology of conceptual representations in unsupervised generative models. In: 26th International Conference on Information Society and University Studies, Kaunas, Lithuania (2021).
9. Dolgikh, S.: Categorized representations and general learning. In: 10th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions (ICSCCW-2019) Prague Czech Republic. Advances in Intelligent Systems and Computing Springer, Cham 1095 93–100 (2019).
10. Gondara, L.: Medical image denoising using convolutional denoising autoencoders, in: 16th IEEE International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 2016, 241–246.
11. A P S.C., Lauly S., Larochelle H., Khapra M.M., Ravindran B. et al.: An autoencoder approach to learning bilingual word representations. In: 27th International Conference on Neural Information Processing Systems (NIPS'14), Montreal, Canada 2, 1853–1861 (2014).
12. Rodriguez, R.C., Alaniz, S., and Akata, Z.: Modeling conceptual understanding in image reference games. In: Advances in Neural Information Processing Systems (Vancouver), 13155–13165 (2019).
13. Yoshida, T., Ohki, K.: Natural images are reliably represented by sparse and variable populations of neurons in visual cortex. Nature Communications 11, 872 (2020).
14. Bao, X., Gjorgiea, E., Shanahan, L.K. et al.: Grid-like neural representations support olfactory navigation of a two-dimensional odor space. Neuron 102 (5), 1066–1075 (2019).
15. Le, Q.V.: A tutorial on deep learning: autoencoders, convolutional neural networks and recurrent neural networks. Stanford University, 2015.

16. Zhou C. and Paffenroth R.C.: Anomaly detection with robust deep autoencoders. In: 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, Canada, 665–674 (2017).
17. Keras: Python deep learning library. https://keras.io/, last accessed: 2021/11/21.
18. LeCun Y.: The MNIST database of handwritten digits. Courant Institute, NYU Corinna Cortes, Google Labs, New York Christopher J.C. Burges, Microsoft Research, Redmond (2007).
19. Fukunaga, K., Hostetler, L.D.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory 21 (1), 32–40 (1975).
20. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 226–231 (1996).
21. Hassabis D., Kumaran D., Summerfield C., Botvinick M.: Neuroscience inspired Artificial Intelligence. Neuron 95(2), 245–258 (2017).
22. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175–185 (1992).
23. Liou, D.-R., Liou, J.-W., Liou, C.-Y.: Learning behaviors of perceptron. iConcept Press ISBN 978-1-477554-73-9 (2013).
24. Schölkopf, B., Smola, A. J.: Learning with Kernels. Cambridge, MA MIT Press ISBN 0-262-19475-9 (2002).
25. Garm, A., Poussart, Y., Parkefelt, L., Ekström, P., Nilsson, D-E.: The ring nerve of the box jellyfish Tripedalia cystophora. Cell and Tissue Research 329 (1), 147–157 (2007).
26. Roth G, Dicke U.: Evolution of the brain and intelligence. Trends in Cognitive Science 9 (5), 250 (2005).
27. Dolgikh, S.: Synchronized conceptual representations in unsupervised generative learning. In: 13th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2021), Mirlabs (2021).