Image clustering by autoencoders

A S Kovalenko[1], Y M Demyanenko[1]

[1]Institute of mathematics, mechanics and computer Sciences named after I.I. Vorovich, Milchakova street, 8a, Rostov-on-Don, Russia, 344090

e-mail: alexey.s.russ@mail.ru, demyanam@gmail.com

Abstract. This paper describes an approach to solving the problem of finding similar images by visual similarity using neural networks on previously unmarked data. We propose to build special architecture of the neural network - autoencoder, through which high-level features are extracted from images. The search for the nearest elements is realized by the Euclidean metric in the generated feature space, after a preliminary decomposition into two-dimensional space. Proposed approach of generate feature space can be applied to the classification task using pre-clustering.

1. Introduction

Nowadays there are a large number of approaches to solving the classification problem [1]. But as a rule, they all resolve into the use of a model class with a teacher-learning-based algorithm. All of them are united by one major drawback - the requirement of marked data for training. When a new task arises in the field of computer vision, as a rule, the marked data is missing, and it is necessary to spend money on their marking.

If we consider approaches based on methods of uncontrolled learning, such as clustering algorithms, they are usually focused on working with data of small dimensions. If we consider images as processed data, they usually have a high dimension. Lowering the dimension of space, for example, by the method of principal components, still does not give space to which clustering algorithms can be effectively applied.

There is a need to build a map acting from the image space Ω (1) into a certain feature space of these images, to which you can effectively apply decomposition methods and directly produce clustering and search for nearby objects by visual component.

2. Latest work

The most frequently used approach to solving the problem of reducing the dimension is the method of principal components. But it can be applied only to rectilinear data. If we consider objects of large sizes, the probability of their good separation becomes small. But if they constitute a mixture of objects belonging to normal distributions with different parameters, they can be separated using the t-SNE algorithm (Laurens van der Maaten Visualizing Data using t-SNE) [2]. In most cases, there is work with data that does not meet these requirements. There is a need to build a mapping from the current space of objects into the space of their descriptive features, which will be imposed the requirement of their distribution under the normal law. Such a problem is considered in the paper on variational autoencoders (Doersch C. Tutorial on Variational Autoencoders) [3].

In our work, for this purpose an encoder was built and trained, which is the necessary mapping into the space of attributes of images distributed according to the normal law. Further, the t-SNE algorithm can be applied to the resulting space to further decompose the space into dimensions, where clustering algorithms will work well.

3. Building of encoder

For training and testing of models, a set of images "MNIST", which contains the images collection of handwritten numbers. An example of the data is shown in the figure 1.

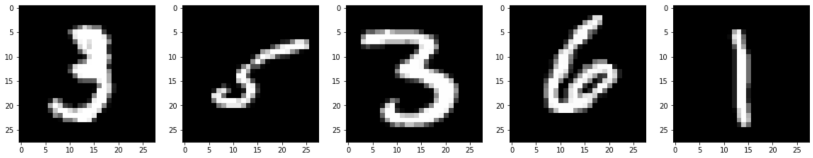

Figure 1. MNIST example.

Autoencoder was built with the architecture shown in the figure 2. For the general concept of architecture in more detail see [3].

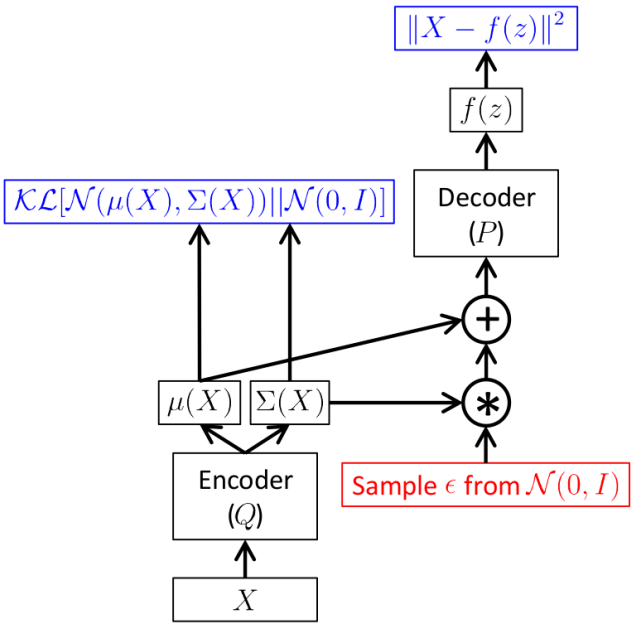

Figure 2. General block diagram of the auto encoder model, where the blue blocks indicate error functions.

During the experiments, the following architectures of the neural networks of the encoder (Q) and decoder (P) networks, built only on fully connected layers and convolutional networks, were used. The parameter for these networks was the dimension of the hidden space.

A detailed description of the architectures of these models in the form of flowcharts is in the repository attached to the work. A brief description of the models is shown in the pictures 3 and 4.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_15 (InputLayer)        (None, 28, 28, 1)         0
_________________________________________________________________
Encoder (Model)              (None, 10)                237972
_________________________________________________________________
Decoder (Model)              (None, 28, 28, 1)         237456
=================================================================
Total params: 475,428
Trainable params: 473,892
Non-trainable params: 1,536
_________________________________________________________________
```

Figure 3. Summary for a model built on fully connected layers.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_9 (InputLayer)         (None, 28, 28, 1)         0
_________________________________________________________________
Encoder (Model)              (None, 10)                25385
_________________________________________________________________
Decoder (Model)              (None, 28, 28, 1)         24924
=================================================================
Total params: 50,309
Trainable params: 50,309
Non-trainable params: 0
_________________________________________________________________
```

Figure 4. Summary for a model built on convolutional layers.

Both models were trained on the above 60000 data set. The Adam [4] optimization algorithm was used for training. Number of learning epochs is 500. Keras framework: keras with backend tensorflow was used for building and learning.

At the output, the encoder returns two vectors: the vector of the mean value for the distribution to which the object belongs and the vector of the covariance matrix in a diagonal form. To construct the set H, we use the average value, since it characterizes the cluster centroid in the space of hidden features, where Ω - original objects set (1), g - encoder.

$$\Omega = \{I_m\}_{m=1}^{N} \tag{1}$$

After training encoder, we construct the desired set H as follows:

$$H = \{g(x)|x \in \Omega\}. \tag{2}$$

Now we lower the dimension of the set H using the algorithm of distributed stochastic selection of neighbors (t-SNE):

$$\hat{H} = \text{t-SNE}(H),\ \forall h \in \hat{H} \Rightarrow dim(h) = 2. \tag{3}$$

On the other hand, you can set the dimension of the hidden space of the variational auto encoder, equal to 2, and then we get a set of objects of the desired dimension. But in this case, the loss of information increases when the object is encoded by an encoder, and the results of decomposition with this approach are worse.

4. Experiments

Was took 5000 elements from the " MNIST " data set and under exposure of the encoder using the t-SNE algorithm, we get the set $\hat{H}$ (3). Consider the examples of $\hat{H}$ sets obtained using the g encoder model with various parameters of the hidden space size. Figure 5 depicts the set $\hat{H}$ when using the dimension of the hidden space equal to 2, without further using t-SNE.

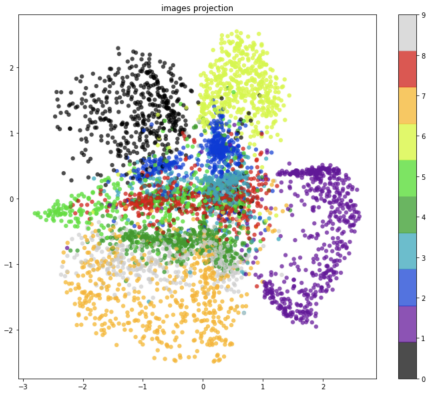

Figure 5. Visualization of the hidden space of dimension 2, obtained by a fully connected encoder.

Figure 6 depicts the set $\hat{H}$ constructed using a fully meshed model with a hidden space dimension parameter of 10 using the t-SNE algorithm.

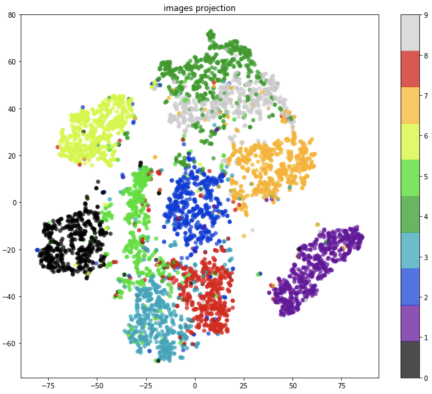

Figure 6. Hidden space of dimension 10, obtained by a fully connected encoder, followed by t-SNE.

Figure 7 depicts the set $\hat{H}$ constructed using the convolutional model with the parameter of the dimension of the hidden space equal to 10 using the t-SNE algorithm.

It can be observed that when using the hidden space of a higher dimension and the subsequent action on it by the t-SNE algorithm, the classes become better separable, picture 6. This is due to the fact that the auto encoder better restores the image at the output and, as a result, the space of hidden features becomes more representative. When using convolutional architecture,

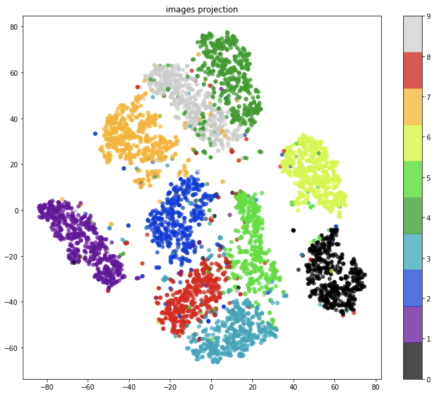

Figure 7. Hidden space of dimension 10, obtained by a convolutional encoder, followed by t-SNE.

visual separability of classes shows similar results, picture 7, and the number of parameters for this architecture is an order of magnitude smaller than that of the whole consisting of fully connected layers. Since when training a variational autoencoder, the encoder tends to predict the parameters of the normal distribution to which the object should belong, then when considering the set of predicted average values, we get the space of normally distributed values. The t-SNE algorithm uses the proximity metric of objects in the normal distribution, which results in a good decomposition result. There is also a method for reducing the dimension of space, based on the selection of the main components (PCA) [5], but it shows the worst results when applied to this set of hidden features, picture 8.

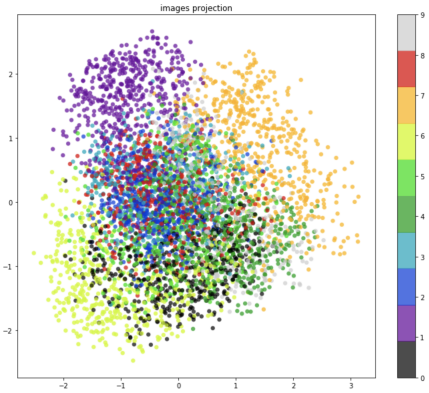

Figure 8. Visualization of the space obtained by the action of the encoder consisting of fully connected layers, after decomposition by PCA.

4.1. Search of nearest images and clustering

Random elements from the set $\hat{H}$ (3) are taken as an example of searching for the nearest images by visual similarity, for which 5 nearest elements from the same set are found using the Euclidean

metric, pictures 9 and 10.

Since in the set $\hat{H}$ (3) the data clusters have a random non-linear form, the clustering algorithm DBSCAN [6] is applied. After splitting the set into clusters, 10 representatives of each class are taken and the voting method selected labels for each class from the available data markup. After that, the classification accuracy is evaluated. It is 82.2%. If you reduce the space to dimension 2 only with an encoder, the accuracy is 75.9%.

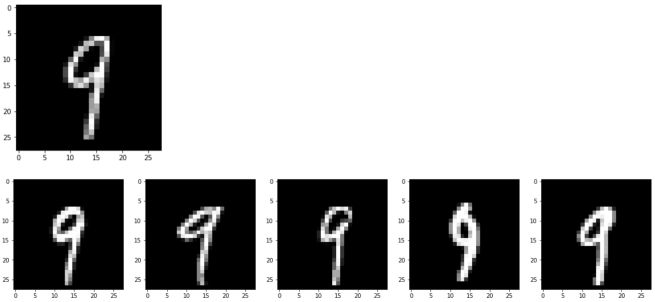

Figure 9. Original image (top) and 5 closest (bottom).

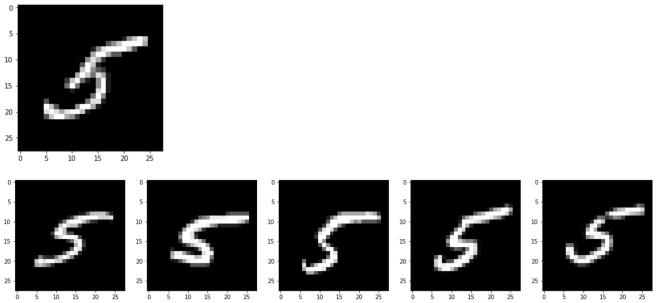

Figure 10. Original image (top) and 5 closest (bottom).

5. Results

In this work, a variational auto encoder model was built, trained on the MNIST task data set. The experiment shows that the described approach of using a larger dimension of the hidden space with its further decomposition using the t-SNE method gives better separability of classes compared to reducing the dimension only by the auto encoder, and gives higher accuracy when clustering the set, 82.2% instead of 75.9%.

6. Conclusion

The considered approach allows us to solve the classification problem on previously unallocated data and search for the closest ones based on visual similarity.

You can also select the nearest elements to the cluster centroids, which will be more likely to be correctly classified during clustering, and train the classifier based on the [7] neural network, which may allow you to classify all the input data with better precision.

7. Applications

Link to GitHub repository with implementation: Clustering-by-VAE

8. References

[1] Kotsiantis S 2007 *Supervised Machine Learning: A Review of Classification Techniques* (Peloponnese: Department of Computer Science and Technology University of Peloponnese) 249-268
[2] Maaten L 2008 Visualizing Data using t-SNE *Journal of Machine Learning Research* **9** 2579-2605
[3] Doersch C 2016 Tutorial on Variational Autoencoders *CoRR* ArXiv: abs/1606.05908 (28.05.2019)
[4] Kingma D 2015 Adam: A method for stochastic optimization *Scottsdale: ICLR* ArXiv: abs/1412.6980 (28.05.2019)
[5] Shlens J 2014 A Tutorial on Principal Component Analysis *CoRR* ArXiv: abs/1404.1100 (28.05.2019)
[6] Ester M 1996 *Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise* (AAAI Press) 226-231
[7] Wu H 2017 *CNN-Based Recognition of Handwritten Digits in MNIST Database* (Berlin: Springer)