Analysis of the latent space of pre-trained deep convolutional neural networks in the problem of automatic segmentation of color images

V Galkin¹, A Makarenko¹ and D Targamadze²

¹Laboratory of Computational Cybernetics, V A Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya street, 117997, Moscow, Russian Federation
²Department “Robotic and intelligent systems”, Moscow Aviation Institute (National Research University), 4 Volokolamskoe shosse, 125993, Moscow, Russian Federation

E-mail: galkinv@ipu.ru

Abstract. The paper presents a primary study of the latent space structure of neural networks trained for semantic segmentation. Segmentation was performed in a controlled environment of three classes of colored rectangular shapes. The classic autoencoder and U-net like architectures were chosen as reference architectures. To study the structure of the space, a combination of a perceptron that linearly separates classes and the compression algorithms UMAP and PCA was used. As a result, a tool was obtained for evaluating the quality of a neural network based on the degree of separability of classes in the latent space of the network.

1. Introduction

The task of automatic segmentation is quality popular and is considered to be one of computer vision’s main tasks. It involves splitting digital images, video frames into segments – sets of pixels that belong to a particular object. Segmentation is widely used in such tasks as autonomous driving of vehicles, analysis of medical images, and various national economic and defense applications. The task of semantic segmentation is closely related to other tasks, such as: detection, recognition, identification and diagnostics of observed objects. Of particular interest is the problem solution provided that there is a high level of noise, the presence of masked (poorly observed) and/or overlapping objects in the field of view.

Recently, deep neural networks are usually used to solve this class of problems, the success of which lies in the possibility of forming low-level spatial representations, as well as in the adaptability and ability to function with wide variations of the phonon target and noise environments. A detailed description of the evolution of the neural network approach to solving the segmentation problem is described in the review papers [1-3].

To find effective neural networks architectures that solve various problems, including segmentation, as well as to evaluate the quality of network operation, it is necessary to conduct a study of the latent space. A latent space is a compressed, relevant representation of data that occurs when the source data is transformed into a low dimensional space that occurs when data passes through the network layers.
The latent space of a neural network is closely related to the concept of effective subspaces or effective manifolds in data. During the investigation [4] of the effective manifold in the data of sounds produced by various living creatures by isolating the effective manifold from sound tracks through nonlinear projection UMAP [5] with the preservation of the global structure and further studying the possibility of clustering. So, in study [6], the authors demonstrated that natural data is concentrated close to a low dimensional manifold and experimentally showed that the success if deep learning is potentially associated with a low-dimensional effective manifold. In addition [7], the research of the latent space of 3-d objects was conducted, during which the authors demonstrated the fact that during training, the neural network approximates the effective manifold contained in the data.

Finally, in the mention [8,9] analyzes the latent space of small-sized networks for the problem of classification of two geometric shapes. From the above results, we can conclude that the neural network transforms the source data in such a way that the classes in the latent space become linearly separable. Moreover, if the size of the latent space is smaller than the dimension of the effective data manifold, then the classes in the latent space are not separable, but the network statistically tries to reduce the inseparable part. Thus, the question arises of choosing the minimum size of the latent space in which the classes are linearly separable. Moreover, the minimum value will be optimal, since additional, uninformative dimensions are introduced only by noise, which affects the accuracy of the network. It should also be noted that the author does not introduce a qualitative assessment of the linear separability of classes, but shows it only graphically.

2. Experimental part
In this paper, we will propose an approach for evaluating the performance of a neural network that automatically captures a set of qualitative characteristics. The approach is based on the estimation of the degree of separation of recognized objects classes (classification/segmentation task) in the latent space of the neural network.

Our proposed approach for evaluating the quality of the model is to use a single-layer perceptron that separates data linearly, and a combination of perceptron and data compression algorithms to assess the effect of the dimension of the latent space on the ability of the network to divide objects of classes linearly. As data compression algorithms the following was used:

1. UMAP – nonlinear compression with global data structure preservation.
2. PCA – linear compression of data while preserving the maximum possible amount of relevant information.

The scheme of implementation of the approach is as follows: sequential compression (starting from 0) over a certain grid in the dimension space with the use of UMAP and PCA algorithms and the usage of perceptron to the formed marked-up latent space.

Thus, the dependence of the degree of objects separability in the latent space of the pre-trained neural network on the dimension of the latent space is formed. From the obtained results it is possible to observe the quality of the linear separability of the classes (presumably in the best case, when the network is "perfect the classes are strictly linearly separable) and find the optimal size of the latent space of the network, since the redundancy of the dimension of the latent space of the network, given a priori, introduces noise interference (noise dimensions) and probably leads to a decrease in the accuracy of the network.

As a test task to demonstrate the method, we will choose the segmentation task in a managed environment.

2.1. Problem statement
The task we are solving is a three-class semantic segmentation of rectangular multi-colored objects in the image. The input data is three-channel synthetic images of intersecting rectangles of different color ranges, and the resulting values are semantic masks showing the ratio of pixels to a certain class. The central approach in this paper is deep learning, which consists in learning a deep convolutional neural network to solve the task. The problem of semantic segmentation is well studied and can be solved with
high accuracy by neural networks, so the task of rectangular multi-colored objects segmentation is not of scientific interest, but in this paper, it acts as an auxiliary task. The main task is to study the structure and properties of the latent (hidden) space of a neural network trained on the segmentation problem.

2.2. Data set

2.2.1. Input data. The input data is a three-channel RGB image with a size of $64 \times 64 \times 3$. The images themselves show rectangles with the color parameter $\delta$, which characterizes the length of the color range, as indicated in table 1. The color of the rectangle for each class is chosen randomly according to the uniform distribution law from the presented ranges. Thus, each image contains three rectangles – representatives of the class.

**Table 1.** Class color ranges, $\delta$ – set parameter ($\delta = 15$).

|     | I               | II              | III              |
|-----|-----------------|-----------------|-----------------|
| R   | $[42 - \delta, 42 + \delta]$ | $[127 - \delta, 127 + \delta]$ | $[212 - \delta, 212 + \delta]$ |
| G   | $[127 - \delta, 127 + \delta]$ | $[212 - \delta, 212 + \delta]$ | $[42 - \delta, 42 + \delta]$ |
| B   | $[212 - \delta, 212 + \delta]$ | $[42 - \delta, 42 + \delta]$ | $[127 - \delta, 127 + \delta]$ |

To draw a rectangle, the following parameters are needed to set:

1. The center of the rectangle $(x, y)$ in the image plane, which is randomly selected according to the uniform discrete distribution law $U[0, 64]$.
2. Rectangle dimensions $(h, l)$, selected from the same distribution as the center $(x, y)$.

Finally, it is necessary to say that the order of generating rectangles from three classes in one image is also random. It is done so in order to give uniformity to the variability of overlapping rectangles on each other.

Thus, to set a rectangle, you need 7 parameters, in other words, an instance of the class has 7 degrees of freedom (variability), namely:

1. Three coordinates that define the color of the rectangle in the RGB color space.
2. Two coordinates that define the center of the rectangle in the image area.
3. Two parameters that characterize the size of the rectangle.

Moreover, for the last two degrees of freedom, the equivalence with respect to the rotation of the image by 90 degrees is valid. And the classes of objects that appear in images are isotropic.

2.3. Getting pre-trained networks

2.3.1. Used neural networks. For the solution, three neural networks were formed, which differ in the volume of parameters, the number of hidden layers, as well as topology.

The architectures used and the appropriate notations for them are introduced below:

1. $\text{NN}_1$ – convolutional auto-encoder consisting of a sequential stack of convolutional and pooling layers.
2. $\text{NN}_2$ – U-net-loke architecture, characterized by a smaller width, depth, and capacity;
3. $\text{NN}_3$ – U-net-like architecture is identical to $\text{NN}_2$ in which ordinary convolutions (Conv2D) are replaced by Separated Convolutions;

Comparative characteristics are shown in table 2.
Table 2. Comparative characteristics of neural networks.

|                          | NN₁        | NN₂        | NN₃        |
|--------------------------|------------|------------|------------|
| Number of Trainable     | 21 356     | 62 264     | 11 063     |
| parameters              |            |            |            |
| Network Depth            | 13         | 24         | 24         |
| Network Width            | 32         | 64         | 64         |
| Network Input Size       | 64 × 64 × 3| 64 × 64 × 3| 64 × 64 × 3|
| Network Output Size      | 64 × 64 × 4| 64 × 64 × 4| 64 × 64 × 4|
| The layer after which    | 6          | 10         | 10         |
| the latent space is      |            |            |            |
| formed                   |            |            |            |
| Latent Space Size        | 2 048      | 1 024      | 1 024      |

The implementation of artificial neural networks was carried out in the keras framework (version 2.2.4) with the tensorflow backend (version 1.13.1).

2.3.2. Training. To train neural networks, a training data set was generated, the total size of which is 200 000 samples. For testing networks, a separate data set was created, the size of which is 15 000 samples.

For training, as a loss function, a function based on the dice coefficient was chosen.

The chosen network training strategy is the same for all and is formed as follows:
1. Training for 25 epochs with a batch size of 2048 on the Nadam optimizer, the default settings of the optimizer.
2. Additional training of a model with a duration of 50 epochs with a batch size of 2048 on the SGD optimizer, the initial learning rate is 0.02, which decreases exponentially with a damping parameter of 0.02/50.

2.3.3. Results. In this section, we present the results of computational experiments, the conditions of which are described in sections 2.3.1, 2.3.2. To evaluate the quality of networks, the following metrics will be used on the test data set: Dice score, Jaccard index.

Table 3. Comparison of the accuracy of trained networks on a test dataset. Format in the table [Dice score, Jaccard index].

|                          | NN₁        | NN₂        | NN₃        |
|--------------------------|------------|------------|------------|
|                          | [0.996, 0.989] | [0.999, 0.999] | [0.999, 0.999] |

The results of the experiments are shown in table 3. Predictably, neural networks cope with the task with high accuracy.
2.4. **Latent space study**

2.4.1. **Auxiliary data set.** To study the cluster structure, additional images that are similar to the images in Section 2.2.1 are needed to be created, but they contain only one class, a certain class of rectangle. The background class will be excluded. Such images determine the position of the class, with certain rectangle parameters, in the hidden space. Due to the "artificiality" of the data set and the flexibility of the developed software, the generation of such a data set is not difficult.

The class-defining parameter is the color of the rectangle, so when studying the structure of the latent space, it is necessary to focus on this parameter of variability. Because of this, we build a research strategy as follows: we fix all the parameters of variability in the rectangle (for all three classes) and explore such a latent space, gradually increasing the variability of the input data, meaning reducing the number of fixed parameters.

Data sets were formed where variable parameters (i.e. free, unfixed) are:

1. rectangle color (3 parameters), the rest are fixed \((x, y, h, l) = (32, 32, 10, 10)\);
2. the color of the rectangle and width, fixed \((x, y, l) = (32, 32, 10)\);
3. the color of the rectangle and the center on the ordinate axis, fixed \((x, h, l) = (32, 10, 10)\);
4. color, length and width of the rectangle, fixed \((x, y) = (32, 32)\);
5. the color and position of the center, fixed \((h, l) = (10, 10)\);
6. all rectangle parameters are free.

In all of the above data sets, there are 5 000 samples per class, so one set contains 15 000 images. The corresponding class labels to the images are saved for later use. For the study of latent spaces, we will use all three networks. For the convenience of further notation, we introduce the following notation, if the text uses the notation “data set No.\(i\),” where \(i = \{1, 2, 3, 4, 5, 6\}\), then the above numbering of data sets is meant.

2.4.2. **The degree of separability study.** To study the degree of separability, we use the tool described in section 2. The configuration of the computational experiment is as follows.

A learning strategy was chosen to train the perceptron:

1. Nadam optimizer with standard parameters for 35 epochs with a batch size of 256;
2. Additional training using the SGD optimizer with an initial learning rate of 0.02 and a damping parameter 0.01/25 for 25 epochs with a batch size of 256.

To form the dependence of the degree of separability on the dimension of data compression, a grid was used, the nodes of which are powers of the number 2 to the full dimension of the formed latent space. For each node of the network, the perceptron was re-trained “from scratch” and the best accuracy was selected according to the \(F_1\) score. The results obtained are shown in figures 1-6.
Figure 1. Changing the separability of classes from the compression ratio of the latent space of the network $\text{NN}_1$ by the UMAP algorithm.

Figure 2. Changing the separability of classes from the compression ratio of the latent space of the network $\text{NN}_1$ by the PCA algorithm.
Figure 3. Changing the separability of classes from the compression ratio of the latent space of the network NN$_2$ by the UMAP algorithm.

Figure 4. Changing the separability of classes from the compression ratio of the latent space of the network NN$_2$ by the PCA algorithm.
3. Results and discussion

In general, for pre-trained neural networks, a high degree of class separability is observed (maximum accuracy 0.99, 0.98, 0.965 for the used networks, respectively, and data set No 6), which is consistent with the high accuracy of the networks. There is also a tendency to monotonically decrease the degree of separability with a decrease in the dimension into which the latent space is compressed. For low-dimensional representations of the latent space, compression by the UMAP algorithm shows higher values of linear separability of classes (on average by 25 percent), which corresponds to the preservation of the global data structure when applying this algorithm. When using the UMAP
algorithm, abnormal outliers are observed (a sharp decrease in the degree of separability, the level of which returns to the previous one at the next compression), probably associated with the suboptimal selection of hyperparameters of the compression algorithm. Selecting a set of hyperparameters for the UMAP algorithm is a separate, complex task.

It can be noted that for the neural network $NN_1$, in contrast to other networks used, for uncompressed latent space, the level of linear separability is lower than for compression. This potentially indicates that the size of the hidden space chosen a priori for the network is excessive, due to the fact that the capacity of the network with Separable convolutions is greater than that of the neural network $NN_2$. This follows from the fact that a Separable convolution is a sequence of two convolutions, and as is known from the approximation properties of neural networks, an increase in depth leads to an exponential increase in the capacity of the model.

It should be noted that the conducted experiments are consistent with the theoretical studies in [8, 9], in which visual analysis was used as an analysis of the neural network, and the initial data were low-dimensional, and the neural networks were the simplest MLP network. Moreover, the problem of classification was being solved.

4. Conclusion
A primary study was made of the latent space of convolutional neural networks pre-trained on the semantic segmentation problem, using a combination of a perceptron and data compression algorithms. As a result, a tool was developed for evaluating the quality characteristics of neural networks based on the degree of separability of classes in the latent space.

It should be noted that the presented approach does not exclude shortcomings and requires additional improvements both in the theoretical aspect and in the technical implementation. This article is rather staged in nature and does not claim to be used as a final tool. Further research will be carried out in the direction of improving the statistical stability of the approach, eliminating anomalous outliers and refining the theoretical concept, as well as integrating the resulting tool into other tasks, such as the identification problem using metric training [10].

Reference
[1] Sultana F, Sufian A and Dutta P 2020 Evolution of image segmentation using deep convolutional neural network: A survey. Knowledge-Based Systems 201-202 106062 doi: 10.1016/j.knosys.2020.106062
[2] Minaee S, Boykov Y, Porikli F, Plaza A, Kehtarnavaz N and Terzopoulos D 2020 Image segmentation using deep learning: A survey. arXiv preprint arXiv:2001.05566
[3] Bashar A 2019 Survey on evolving deep learning neural network architectures. J. AICN 1(2) 73 doi: 10.36548/jaicn.2019.2.003
[4] Sainburg T, Thielk M and Gentner T 2019 Latent space visualization, characterization, and generation of diverse vocal communication signals. bioRxiv:870311 doi: org/10.1101/870311
[5] McInnes L, Healy J and Melville J 2018 Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426
[6] Na L, Dongsheng A, Yang G, Kehua S, Shixia L, Zhongxuan L, Shing-Tung Y and Xianfeng G 2020 A geometric understanding of deep learning. Engineering 6(3) 361 doi: 10.1016/j.eng.2019.09.010
[7] Wu J, Zhang C, Xue T, Freeman B and Tenenbaum J 2016 Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS 2016) (Barcelona, Spain) p 82
[8] Christopher O 2014 Neural networks, manifolds, and topology, available at: https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/
[9] Hajij M and Istvan K 2020 A topological framework for deep learning. arXiv preprint
[10] Rippel O, Paluri M, Dollar P and Bourdev L 2015 Metric learning with adaptive density discrimination. *arXiv preprint arXiv:1511.05939*