

Object image recognition using multilayer perceptron combined with Singular value decomposition

Vasyl Lytvyn^{1,†}, Ivan Peleshchak^{1,*†}, Roman Peleshchak^{1,†}, Iryna Shakleina^{1,†}, Nazarii Mozol^{1,*†}, Dmytro Svyshch^{1,*†}

¹Lviv Polytechnic National University, 12 Stepana Bandera Street, Lviv, 79013, Ukraine

Abstract

In this paper, a method for recognizing object images with high accuracy using a multilayer perceptron combined with Singular value decomposition (SVD) is developed. The neural network was trained by a back-propagation algorithm using the Adam optimizer based on the Mnist dataset. The singular value decomposition of the matrix was used for data preprocessing and initialization of the network layer weights, which made it possible to increase the image recognition accuracy by 2% with less input data, which is equal to 18% of the data compared to the model without SVD (144px/784px=0.18). In addition, the application of the proposed combined method makes it possible to perform effective short-term training of small neural networks on small photos, unlike existing traditional methods based on VGG and ResNet architectures. The proposed combined method is especially valuable for image recognition in the presence of limited computing resources and training time.

Keywords

multilayer perceptron, singular value decomposition of matrices, object image recognition, initialization of scales, Adam optimizer.

1. Introduction

The task of image recognition is a popular one for all kinds of application purposes. The main problem in this task is that standard methods for training larger neural networks are often not effective when resources are limited or when a given accuracy needs to be achieved quickly. Our work focuses on developing a methodology that combines the SVD method, which preprocesses data and initializes weights, with a neural network to optimize the process of training them on small photos.

The paper emphasizes that the proposed method maximizes performance in the case of short-term training, which is critical when resources are severely limited. Compared to the known neural network architectures (VGG and ResNet), our approach demonstrates a much smaller model size, which is reflected in the computing resource. Compared to using a neural network model without additional processing with SVD, the accuracy of image recognition is lower. The neural network combined with SVD has high performance, which is confirmed by a computer experiment.

The paper discusses in detail the theoretical aspects of SVD and its application in the context of small neural networks and describes the experiments performed and the results obtained. We show how using SVD for layer weights initialization and data preprocessing can significantly improve the efficiency of training in the early stages, giving small networks a significant advantage in the speed and accuracy of image recognition. The final part is devoted to discussing the prospects for further research and the possibility of applying the developed method in various fields.

MoMLeT-2024: 6th International Workshop on Modern Machine Learning Technologies, May, 31 - June, 1, 2024, Lviv-Shatsk, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ vasyly.v.lytvyn@lpnu.ua (V. Lytvyn); ivan.r.peleshchak@lpnu.ua (I. Peleshchak); roman.m.peleshchak@lpnu.ua (R. Peleshchak); ioshakleina@gmail.com (Iryna Shakleina); nazarii_mozol@icloud.com (N. Mozol); svyshch.d.m@gmail.com (D. Svyshch)

ORCID 000-0002-9676-0180 (V. Lytvyn); 0000-0002-7481-8628 (I. Peleshchak); 0000-0002-0536-3252 (R. Peleshchak); 0000-0003-0809-1480 (I. Shakleina); 0009-0003-6770-7609 (N. Mozol); 0009-0004-7882-9676 (D. Svyshch)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The aim of the work is to develop a combination of a multilayer perceptron with the SVD algorithm for recognizing object images with high accuracy (>99%) and a small value of the loss function (<0.1).

2. Literature analysis

In modern data analysis, data classification is one of the most popular tasks solved, in particular with the help of neural networks and the singular value decomposition method.

In the field of pattern recognition, methods that seek to address changes in lighting, pose, and facial expression are key to improving the performance of neural networks. Paper [1] makes a significant contribution to this field by proposing a symmetric singular value decomposition representation (SSVDR) method for image recognition. This method utilizes the singular value decomposition (SVD) and face symmetry to improve the quality of image recognition under varying lighting conditions. The advantage of the proposed method is its ability to create a homogeneous representation of the original image even under large variations in lighting. The simplicity of the method and its high performance with small sample sizes show the possibility of using the method in conditions of limited resources.

In [2], the authors discuss the use of singular value decomposition (SVD) as a means to initialize the parameters of neural networks, as well as an analog of multilayer neural networks in linear algebra. The paper discusses the properties of SVD for linear regression and analysis of overdetermined and underdetermined problems. The paper emphasizes the special usefulness of SVD for generating an initial solution in the optimization of nonlinear networks and the high performance of SVD compared to other initialization methods.

The authors of [3] investigated the problem of classifying brain images using medical image processing methods. In particular, in [3], the authors propose a method that can automatically detect and classify the degrees of brain gliomas. This method combines mutual information with singular value decomposition and automatically selects a set of features to be used in the classifier. The researchers also point out the limitations of their model, noting that MI-ASVD [3] works well only on a specific selected dataset, and may not be effective on others. Therefore, they recommend using different classifiers and expanding datasets to improve classification accuracy in different clinical cases. They also mention the significant requirements for computing resources.

The paper [4] investigates the problem of compression and classification of electrocardiographic (ECG) signals with arrhythmias using the singular value decomposition (SVD) method. It is noted that ECG monitoring systems are widely used in telemedicine applications, where ECG signals are first compressed before transmission and storage. The method of using SVD for decomposition of ECG signals is proposed, after which various classifiers are used, in particular convolutional neural networks (CNN) and support vector machines (SVM). The SVD technique is used to compress and reconstruct the ECG signal, and then the resulting compressed signal is classified and detected using two types of classifiers.

The authors of [5] investigate the application of the singular value decomposition (SVD) method to accelerometer data for recognizing falls and activities using one-dimensional convolutional neural networks (CNNs). Three dimensionality reduction methods - SVD, sparse PCA, and kernel PCA - are compared for their effectiveness in extracting useful features for fall and activity recognition. Experiments conducted on three datasets of falls and activities of daily living show that SVD applied to accelerometer data, together with raw data or acceleration magnitude vector, showed better accuracy in recognizing falls and activities using one-dimensional CNN than other dimensionality reduction methods based on principal component analysis.

Study [6] addresses the problem of efficient initialization of recommender system (RS) algorithms based on singular value decomposition (SVD). The paper solves the problem of developing an efficient initialization method for SVD algorithms. We propose a general framework for initialization using neural embeddings, where a low-complexity probabilistic autoencoder neural network initializes the characteristics of the user and elements. This framework supports explicit and implicit feedback. The article notes that existing SVD algorithms usually initialize the user/item characteristic vectors with random values and thus can achieve local optimization.

When comparing classification approaches using neural networks and the singular value decomposition method, some important characteristics can be identified:

- Neural networks are generally known for their ability to adapt to complex dependencies in data and their ability to use deep learning to automatically identify important features or attributes.

This can lead to high accuracy in classification, especially in cases where the data has a complex structure or non-linear dependencies.

- Neural networks may require a large computational resource to train the model, which is also partially addressed in our work.
- Neural networks, especially deep ones, are often known for their "black box" nature, i.e., difficult to interpret solutions. This can be a problem in some areas where a clear explanatory model is needed. The singular value matrix decomposition method can provide more interpretability because it allows you to isolate the main components of the data. This can help to understand which features or functions are most important in the classification process.
- Singular value decomposition (SVD) is widely used to reduce the dimensionality of data and extract the principal components. This improves the efficiency and speed of data processing. However, SVD can be less effective in detecting complex relationships between features in some data, which can lead to lower classification accuracy, but this problem is solved in our work by combining SVD and fully-connected neural network.

In contrast to the discussed methods, our work focuses on the use of SVD in small fully connected neural networks for classification. We found that using SVD for weight initialization and image preprocessing can improve the accuracy of neural networks with a small number of hyperparameters. Our proposed approach is particularly productive for applications where there are time or computational resource constraints. Our work utilizes SVD to improve performance in small datasets and small images, where neural networks are prone to overfitting. Our research is also aimed at improving the training procedure. Based on computer experiments, we show that the use of SVD in combination with a fully connected neural network has significant advantages for fast training of neural networks with small datasets.

3. Materials and methods

3.1 Definition of SVD

Singular matrix decomposition is a powerful mathematical tool that is widely used in data processing and analysis, including machine learning. This method allows you to decompose an input matrix into three other matrices U, S, and V.T, where U and V.T are orthogonal, and S is a diagonal matrix of singular values.

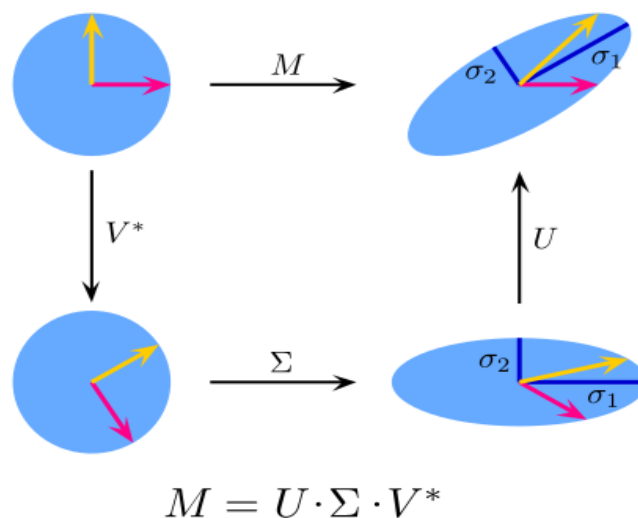


Figure 1: Scheme of the input matrix decomposition

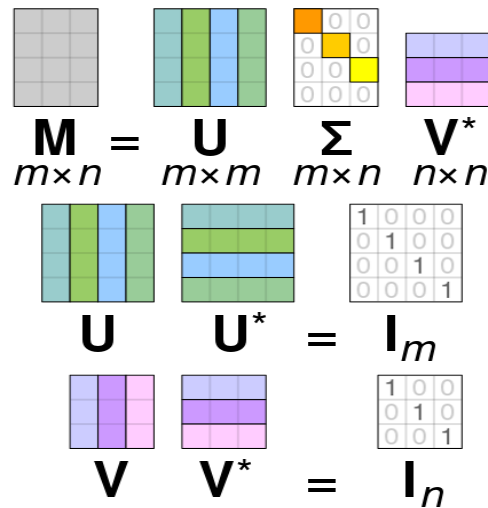


Figure 2: Schematic of the detailed obtaining of each matrix

This decomposition helps to identify the main components of the data matrix, reduce the dimensionality, and initialize the weights in neural networks. An example of the decomposition is shown in Figure 1 and Figure 2.

3.1.1 Using SVD for data pre-processing

In our study, the SVD method is used to optimize data processing. First, the input data is reduced in dimensions for further feeding it to the decomposition function. At the time of SVD, we extract the most important features from the image. By selecting only the first 144 singular vectors, we significantly reduce the dimensionality while preserving important information about the image.

3.1.2. Using SVD to initialize the scales

According to the general idea of the work, it would be inefficient to use all the input data to initialize the weights, since they have already passed the decomposition pre-processing step, so we will focus on which of the decomposition matrices we can use and for what purpose:

1. U-matrix - the left singular vectors of the input matrix that form an orthonormalized basis for the column space. It can be useful for capturing dominant patterns in data, for example, in autoencoder architectures where we are trying to capture variance or data structure in a reduced dimensional space.
2. S-matrix - a diagonal matrix of singular values that show the amount of contribution of each corresponding singular vector to the matrix. Since these values quantify the importance of each singular vector in capturing the variance of the data, initializing the weights with singular values would not be entirely correct and cannot be applied directly, as they do not provide a basis, but rather scale the contribution of each basis vector.
3. V.T-matrix - the right singular vectors of the input matrix form an orthonormalized basis for the row space. It can be particularly effective for layer initialization when you want to project the input data into a space that emphasizes the most significant patterns in the training set.

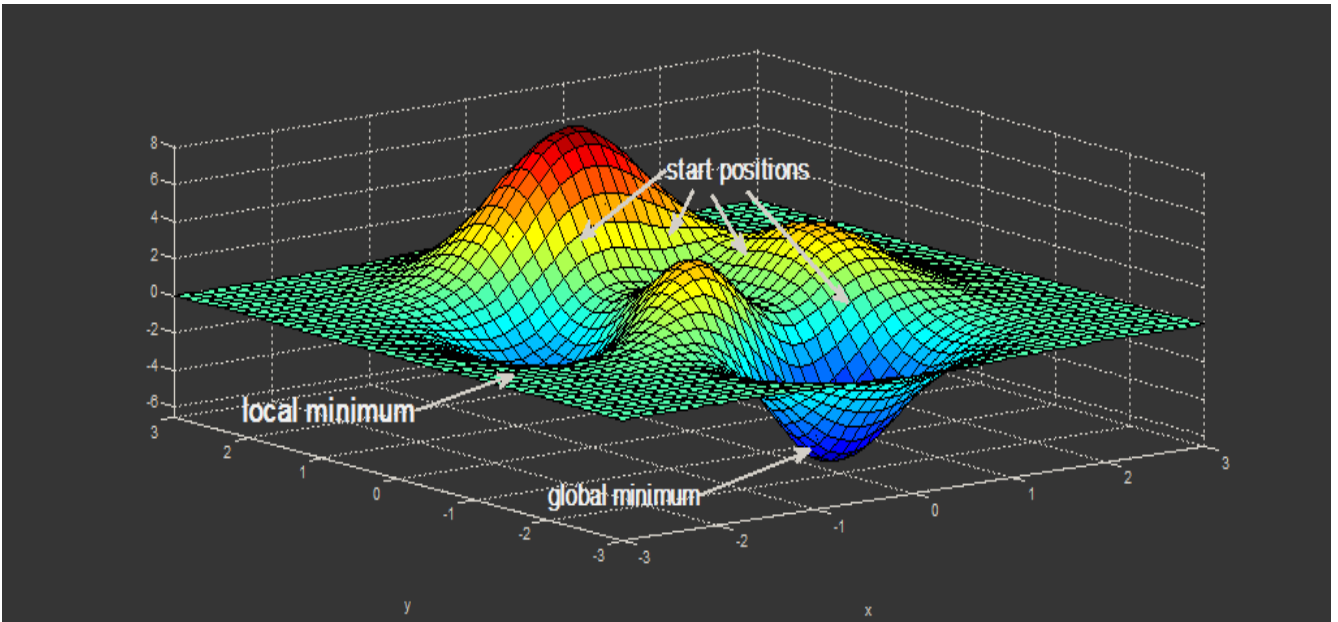


Figure 3: Scheme for obtaining the global minimum of the loss function

Given this data, we use V.T. to extract the most significant patterns from the training data. This will allow us to be closer to the global minimum of the loss function in our case at the beginning of training, which, *ceteris paribus*, will allow us to obtain a higher accuracy result (Figure 3).

3.2 Practical application of SVD

3.2.1 The problem of retraining and the assumption of dimensionality reduction

For the vast majority of small networks, when they are trained for classification, retraining on training data is relevant, which does not allow them to reliably predict test data, since they have a limited number of parameters to generalize the data set, identify the necessary patterns that are present, or overcome the presence of noise. One way to overcome this is to use convolutional neural networks, but our goal is small networks, so this method is not relevant for two reasons:

1. Model size - in both cases, when we learn our feature layers or use ready-made ones (e.g. ResNet architecture or VGG16), we significantly increase the size of the model by memorizing additional parameters, for example, the size of these layers for ResNet starts at 44 mb [7].
2. Training complexity – in the case of using convolutional feature layers (CNNs), we have to train them ourselves, which is time-consuming, and in the case of using ready-made ones, it still leads to a higher load when predicting data, simply because of the larger number of parameters we use for calculation.

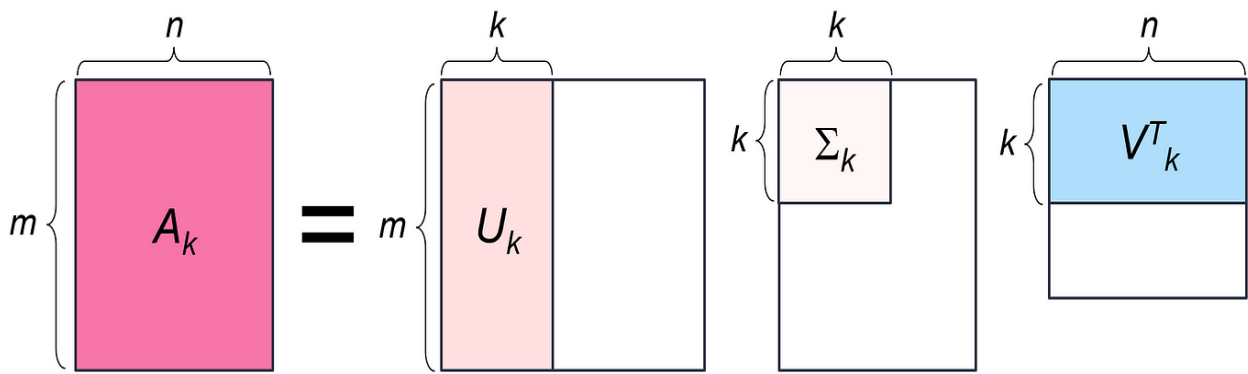


Figure 4: Scheme of taking into account only the most important K features

An alternative to this would be to reduce the dimensionality of the data. But in this case, we will be discarding random pixels, which will lead to significant differences between models under the same conditions. However, it is still feasible to do this with SVD, since we can keep only the components associated with the largest singular values, reducing the feature space to a given number of dimensions (Figure 4). This advantage of the method is the most significant in our work, as it helps to reduce the size of the input data without significant loss of information about it.

4. Preliminary processing and additional information

For this work, we used the Python programming language with the Tensorflow framework and the Mnist dataset. In data preprocessing, augmentation is also used to better generalize the features of the input images (Figure 5).

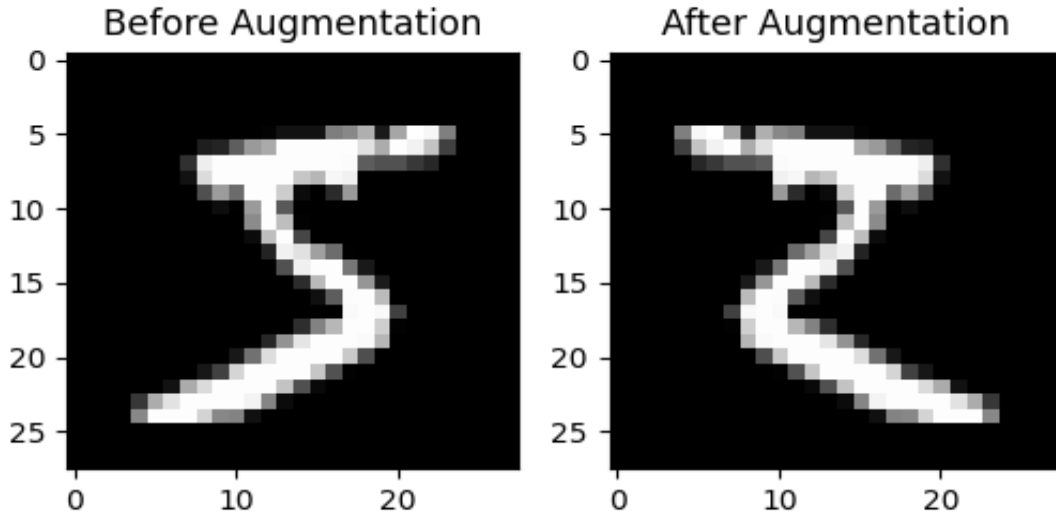


Figure 5: Example of an image before and after augmentation

It is worth noting that different application interpretations of the SVD method are used for weight initialization and image preprocessing. For images, the TruncatedSVD function is used, which differs from the usual method in that it is approximate and adds some noise to the result, which simulates the operation of weak augmentation of the input data. For the weights, an accurate representation of the SVD method is used, since any small change can move the initial training point far away from the global minimum.

5. Mathematical model of classification

Since we use SVD and select only the first N most important features of the image matrix, we take $12 \times 12 = 144$ pixels of the input image ($28 \times 28 = 784$) as N, which is only 18.4% of the input information.

The main task of the model in this paper is multiclass classification.

The task of classification is to find the mapping operator $Y^*: X \rightarrow Y$ for any objects that are not included in the training set with a minimum norm in the Euclidean space:

$$\min \|y^* - y\| \quad (1)$$

where y is the target classifier; y^* - neural network classifier.

The mapping operator is known only on objects of a finite training set: $X_m = \{(x_1, y_1), \dots, (x_m, y_m)\}$, where X_m is the set of training set elements of dimension m. The task is to build an algorithm that can determine whether an arbitrary object $x \hat{=} X$ belongs to the class $y \hat{=} Y$.

5.1 Neural network morphology using SVD

We consider only small feed-forward neural networks, so we chose 1 hidden layer of size 128 with the activation function relu; a flattening layer at the input, and a layer of size 10 with the activation function softmax at the output. The flattening layer has the dimension of the product of all the dimensions of the input photo, and since we get the first N most important features, the dimension is 144. Adam optimizer, categorical cross-entropy loss function, accuracy metric (Figure 6).

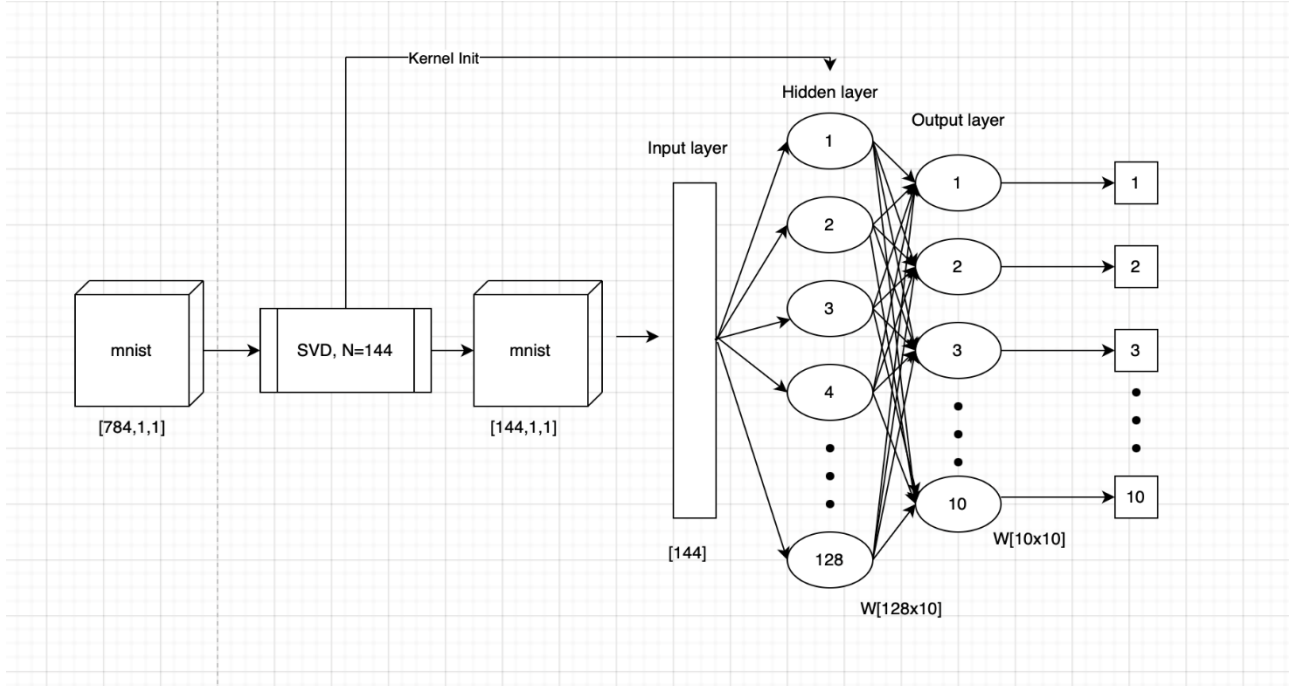


Figure 6: Combined architecture of a neural network with one hidden layer and SVD block

The output of this network will be described by the ratio:

$$y_i = f_{softmax} \left(\sum_{j=1}^{128} \omega_{ij} f_{relu} \left(\sum_{k=1}^{144} \omega_{jk} x_k \right) \right) \text{ where } i \in \{1; 10\}, \quad (2)$$

where $f_{softmax}$ - is the Softmax activation function, f_{relu} is the Relu activation function, ω_{ij} is an element of the weight matrix between the first and second tacked layers, ω_{jk} is an element of the weight matrix between the input layer and the first hidden layer, x_k is an element of the input image vector.

5.2 Neural network morphology without SVD

The structure is completely similar to the first network, except for the dimensionality of the input image and the flattening layer - since we do not discard features here, the entire photo (28×28) is included in

the network, so the dimensionality of the first dimension of the photo and the dimensionality of the flattening layer is 784 (Figure 7).

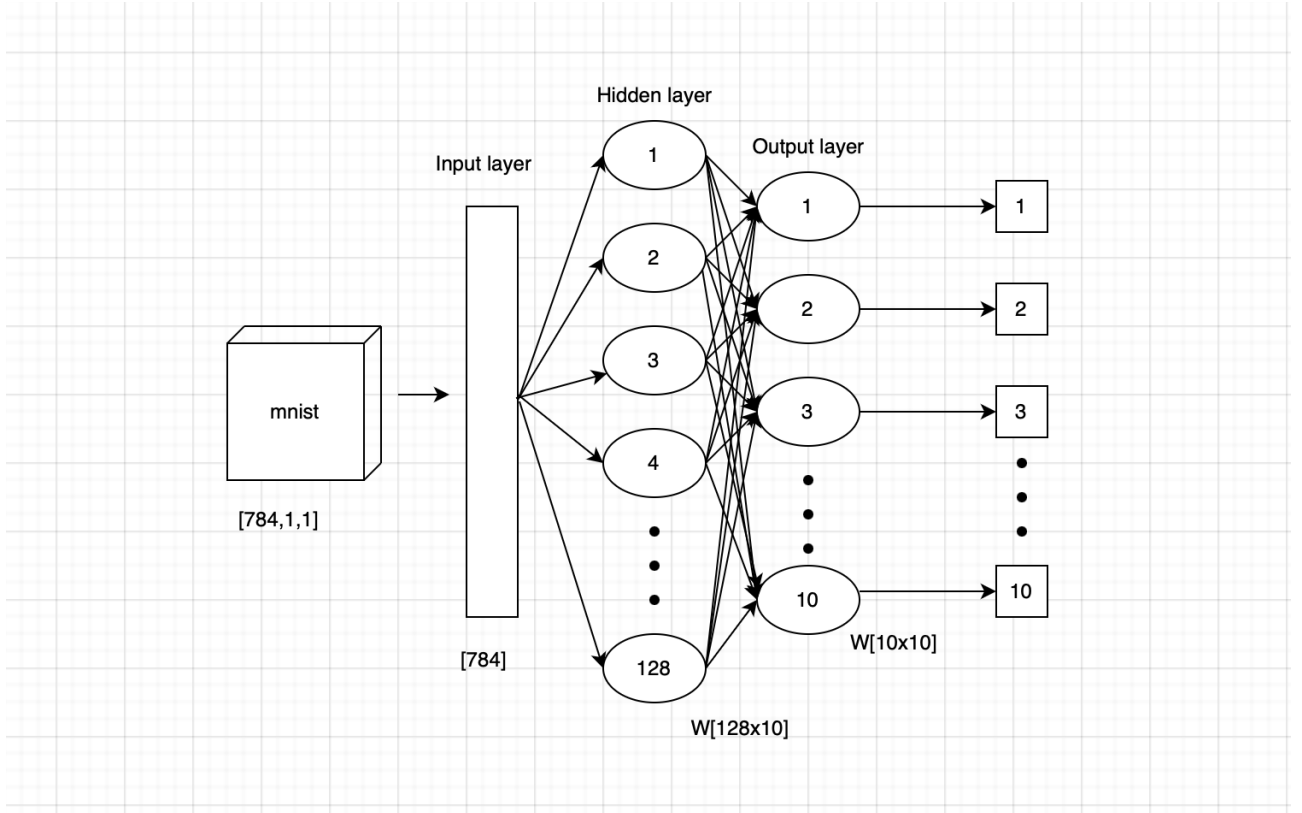


Figure 7: Architecture of a neural network with one hidden layer

$$y_i = f_{softmax} \left(\sum_{j=1}^{128} \omega_{ij} f_{relu} \left(\sum_{k=1}^{784} \omega_{jk} x_k \right) \right) \text{ where } i \in \{1; 10\}, \quad (3)$$

where $f_{softmax}$ – is the Softmax activation function, f_{relu} is the Relu activation function, w_{ij} is an element of the weight matrix between the first and second tacked layers, w_{jk} is an element of the weight matrix between the input layer and the first hidden layer, x_k is an element of the input image vector.

6. Experimental results and their analysis

In this work, we compared the performance of a small fully connected neural network using SVD for image processing and initialization of weights with and without a model. The data for training and testing were obtained from the MNIST dataset, which contains handwritten digits because it contains many small images (28*28 pixels).

An important aspect of the experiment is the analysis of the model's loss function. The results show that when SVD is used for weight initialization and image processing, the loss function quickly stabilizes and reaches its horizontal asymptote, which indicates that it has reached a global minimum, already in the first training epoch (Figure 8). This emphasizes the high efficiency of using SVD in the context of quickly achieving the optimal solution.

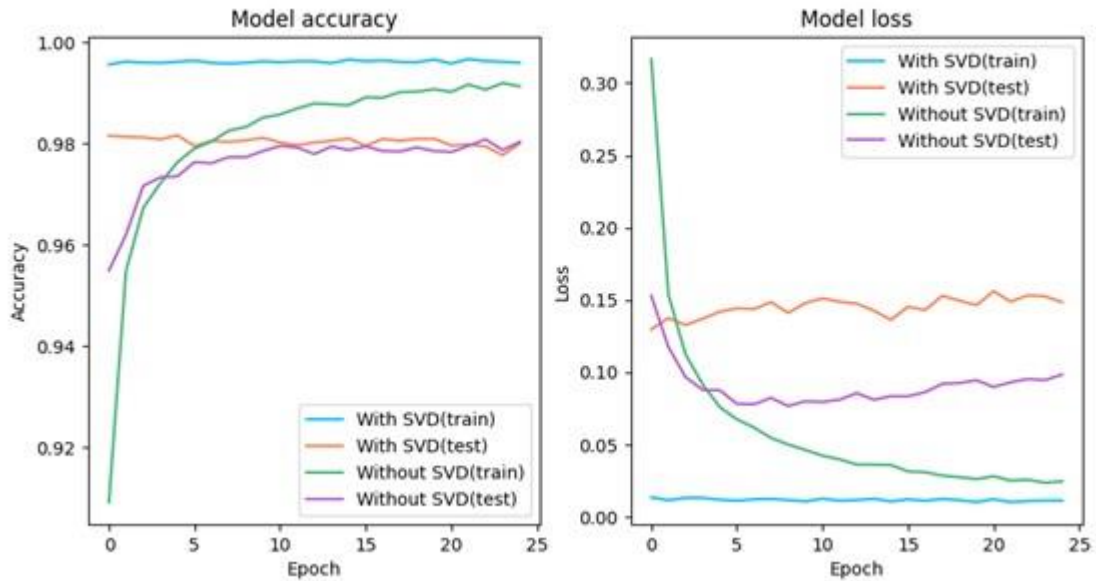


Figure 8: Dependence of image classification accuracy and model loss function on the number of epochs

In contrast, in the non-SVD model, we observe that the loss function decreases gradually over the initial few epochs and continues to show a decreasing trend throughout the training process. Furthermore, even at the end of the training period, for the training sample, the loss rate remains higher than in the SVD model. Concerning the test set, we see a slight increase in losses, but this can be explained by the generally less-than-perfect approximation of the results in our dataset, so we can choose the training set as the key parameter for comparison since the difference in accuracy between the two sets is not significant. This indicates that the model without SVD does not achieve as good a solution as the model with SVD.

The data analysis confirms that the use of SVD for image preprocessing and initialization of weights helps to improve the classification accuracy of small fully connected neural networks on a small number of epochs. The significant reduction of losses in the early stages of training using SVD also emphasizes its potential to accelerate the learning process, making it ideal for scenarios where time is a limited resource.

The results shown in the graphs suggest that the use of SVD can be an important tool for improving the efficiency of small neural networks and can be recommended for a wide range of machine-learning applications.

7. Conclusions

A method for recognizing object images using a multilayer perceptron combined with singular value decomposition (SVD) has been developed. The method allows for efficient image processing in conditions of limited computing resources.

It was found that, as a result, using SVD for initialization of weights and image processing helps to improve the network performance. The use of SVD helps to reach the global minimum faster, which allows for faster achievement of the optimal result, and makes the learning schedule smoother, which simulates a slower learning rate without directly changing it. The use of SVD helps to reach the global minimum faster, which allows to achieve the optimal result faster.

It is proved that the use of SVD improves the accuracy of object image recognition for a small number of training epochs. Experimental results demonstrate that the use of SVD helps to reduce losses in the early stages of training.

The programming language used is Python with the Tensorflow framework and the Mnist dataset. The data preprocessing also uses augmentation to better generalize the features of the input images. Adam optimizer, categorical cross-entropy loss function, accuracy metric. The number of hidden layers

is 1 with a size of 128 with the relu activation function; the input is a flatten layer, and the output is a layer of size 10 with the softmax activation function.

References

- [1] Yuhui Chen, Shuiguang Tong, Feiyun Cong, Jian Xu. "Symmetrical singular value decomposition representation for pattern recognition". *Neurocomputing*. Volume 214, (2016), P. 143-154. DOI: 10.1016/j.neucom.2016.05.075
- [2] Bermeitinger, B., Hrycej, T., Handschuh, S. "Singular Value Decomposition and Neural Networks". *ICANN 2019: Deep Learning*. ICANN 2019. Lecture Notes in Computer Science, vol 11728. Springer, Cham. (2019). https://doi.org/10.1007/978-3-030-30484-3_13
- [3] Z. A. Al-Saffar and T. Yildirim. "A Novel Approach to Improving Brain Image Classification Using Mutual Information-Accelerated Singular Value Decomposition". In *IEEE Access*, vol. 8, (2020), pp. 52575-52587. doi: 10.1109/ACCESS.2020.2980728
- [4] Lijuan Zheng, Zihan Wang, Junqiang Liang, Shifan Luo, Senping Tian. "Effective compression and classification of ECG arrhythmia by singular value decomposition". *Biomedical Engineering Advances*. Volume 2, (2021), 100013. <https://doi.org/10.1016/j.bea.2021.100013>
- [5] H. Cho, S.M. Yoon. "Applying singular value decomposition on accelerometer data for 1D convolutional neural network based fall detection". *Electronic letters*. Volume 55, Issue 6, (2019), P. 320-322. <https://doi.org/10.1049/el.2018.6117>
- [6] T. Huang, R. Zhao, L. Bi, D. Zhang and C. Lu. "Neural Embedding Singular Value Decomposition for Collaborative Filtering". In *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, (2022), pp. 6021-6029. doi: 10.1109/TNNLS.2021.3070853
- [7] Model VGG 16. <https://pytorch.org/vision/main/models/generated/torchvision.models.vgg16.html> . Access date: 08.04.2024