SV-Learn: Learning Matrix Singular Values with Neural Networks

Derek Xu
Ayala High School
derekxu888@gmail.com

William Shiao
Dept. of CSE
UC Riverside
wshia002@ucr.edu

Jia Chen
Dept. of ECE
UC Riverside
jiac@ucr.edu

Evangelos E. Papalexakis
Dept. of CSE
UC Riverside
epapalex@cs.ucr.edu

***Abstract*—The singular value decomposition (SVD) factors a matrix into three separate matrices: two (semi-)unitary matrices whose columns are left/right singular vectors and one diagonal matrix whose diagonal entries are singular values. Typically, performing SVD on big matrices is taxing due to its computational complexity in the cubic order of its dimensions. With the advances and rapid growth of deep learning techniques in a broad spectrum of applications, a fundamental question arises: can deep neural networks learn the singular values of a matrix? To answer this question, we propose a novel algorithm, namely SV-learn, to predict the singular values of a given input matrix by leveraging the advances of neural networks. Numerical results demonstrate that our proposed method outperforms the competing alternatives in terms of achieving lower normalized mean square error on singular value prediction when using real-world datasets. Further, the predicted singular values combined with singular vectors of an input data allow us to reconstruct the input matrices with promising performance.**

I. Introduction

The Singular Value Decomposition (SVD) and eigenvalue decomposition are of prominent importance in broad spectrum of real-world applications such as computer vision [1], signal processing [2], and data science [3], [4], [5]. A key feature of the SVD is that it is *rank-revealing*, which means that it identifies the rank of a given matrix, as well as providing us with a means for evaluating the so-called "low rank" of a given data matrix, which would be a rank typically much smaller than the full rank of the matrix to which we can approximate the matrix with minimal loss of reconstruction accuracy. The concept of that low rank is central to virtually every attempt in using the SVD (or other matrix factorizations) for the purposes of dimensionality reduction, either as a pre-processing step in a analytical pipeline, or as the actual analysis itself (e.g., by identifying latent concepts in term-document matrices [3]). Furthermore, SVD is extremely useful in calculating the pseudo inverse with low-cost.

As a result, the singular value profile of a matrix is an extremely important product of the SVD, since it can characterize a given matrix dataset with respect to the number of latent patterns that it contains. Thus, in this work, we propose a novel neural network based approach that is meant for quick and effective prediction of the singular values of a given matrix. Most of the existing work rely on the direct computation of SVD or eigenvalue decomposition on a matrix or sub-matrix. Recently, a new model was proposed in [6] to predict the latent dimensionality of non-negative matrix factorization using neural networks.

To the best of our knowledge, using deep neural networks to explore the full spectrum of singular values has not been studied yet. Our contributions include:

- Proposing a novel deep learning-based framework to predict singular values of a matrix.
- Evaluating the effectiveness of the proposed method through numerical tests with promising visualized and quantified results.

II. Background

The SVD has been applied successfully to a wide variety of datasets, with applications in image compression [7], [8], recommender systems [9], etc. However, the time complexity of calculating the SVD is significant, making it difficult to scale to extremely large datasets. This has led to work focused on speeding it up for sparse matrices [10], [11], distributing it across multiple machines [12], [13], and increasing its efficiency on accelerators [14].

In this work, we leverage neural networks to help speed up calculating an approximation of the SVD by feeding a model a set of training data to make predictions faster. Neural networks have been shown to be highly effective in learning and modeling non-linear and complex relationships. In our case, we theorize that with proper optimization and training, we can use an neural network to predict a significant amount of the singular values and vectors with less time complexity than the typical SVD.

This application has a major use in big data where execution time is extremely important. Instead of having to calculate the SVD from scratch over all of the matrices, we can use a shared model trained over many sample matrices to decrease the resource usage and run times of future factorization. In conjunction with such a model, it is possible to perform many tasks that are dependent on the SVD, such as principal component analysis, spectral analytics, and manifold embeddings, much faster, especially on larger data sets.

III. Problem Formulation and Proposed Method

A. Problem Formulation

We tackle the problem of predicting the singular values of two-dimensional matrices by training supervised neural networks with matrices and their respective singular value decomposition. To get more accurate results, our model needs to be trained on matrices of varying ranks. There are a couple of ways to do this, but our method involves using real data sets and "windowing" (creating certain sized matrices by moving a window of size $m \times n$ over the data with different dimensions) the data to fit our model. The goal of this process is for a model to successfully predict the singular values of any matrix of the size that it was trained on. This can then be compared to other methods of solving for the decomposition of a matrix such as our SVD-LIGHT in Table 1, which uses a subset of a data set for the singular values and vectors, or the full SVD. This theoretically will decrease run times and resource usage for large data sets where repetitively finding the singular values of matrices is necessary.

Algorithm 1 SVD-LIGHT

Inputs: matrix $\mathbf{M}$; k, the number of rows and columns from $\mathbf{M}$ used for singular vector approximation

Outputs: submatrix's singular vectors stored in the columns of $\mathbf{U}_C$ and $\mathbf{V}_R$; sparsified singular value matrix $\mathbf{\Sigma}_O$

Description: A method of decomposing a matrix where SVD is applied to a smaller subset of a matrix which acts as a form of comparison against using the full SVD and our trained model.

```
procedure SVD-LIGHT(M, k)
    Σ ← PREDICTSVD(M)
    M_R ← SAMPLEROWS(M, k) // Randomly select k rows of M to form a submatrix M_R
    M_C ← SAMPLECOLS(M, k) // Randomly select k columns of M to form a submatrix M_C
    U_R, Σ_R, V_R^T ← SVD(M_R) // Perform SVD on M_R
    U_C, Σ_C, V_C^T ← SVD(M_C) // Perform SVD on M_C
    diag(Σ_O) = [diag(Σ)(1 : k), 0, ..., 0] // Preserve the top k singular values of M and zero-out the remaining ones forming the diagonal matrix Σ_O
    return U_C, Σ_O, V_R^T
end procedure =0
```

B. Proposed Method

Dataset: We opt to use the CIFAR-10 dataset [15] to get 32×32 square matrices in order to obtain the most accurate SVD predictions off of real data. The dataset also allows us to easily provide visualizations for the accuracy of our predictions. We then scaled down the dimensions of each image to create a 2D 32×32 matrix $\mathbf{M}$, which holds the gray-scale values of the image at each pixel within the original image. For our other data set, we opted to use the Spambase [16] dataset, which we obtained from the UCI Machine Learning Repository [17]. We extract 32×32 windows from the feature matrix to match dimensions of the CIFAR-10 set. We chose as an example of tabular data. For all of the matrices from these sets, the SVD was taken to calculate the singular values of the data. We then saved the original matrices, along with their singular values taken from the SVD, to use for training our neural network.

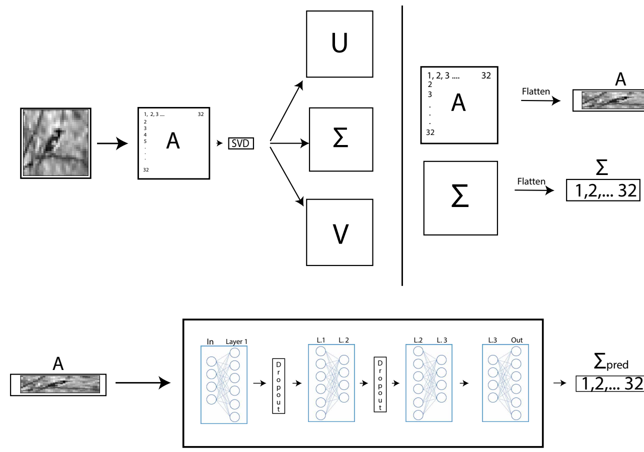


Fig. 1: Overview of the proposed model.

Training neural regression model: Our goal with our neural network is to find the singular values of a given matrix. After getting the singular values from the SVD, we developed a model to train on the data. From our testing, we found that using a regression model with 2 hidden layers with 2 dropout layers worked the best to predict our singular values. To be able to train the neural networks, they require a 1D array for each unit. To achieve this, we flattened our original matrices, giving us a data set full of 1D arrays $\mathbf{m}_f$ and their corresponding 1D arrays of singular values $\mathbf{s}_f$ where f denotes the sample index. The matrices and singular values were then fed into the neural network with $\mathbf{m}_f$ as input and $\mathbf{s}_f$ as the base truth. For training, we used L_1 loss to adjust the weights of the neurons. This yields a model that outputs a 1D array of singular values for $\mathbf{m}_f$; the overview of our proposed framework can be found in Fig. 1. The purpose in doing this is that for wider applications, users will simply need to perform SVD operations only once for a few samples of their datasets. They can use the model that is trained using the SVD from a couple of samples to the entirety of their data, eliminating the need to perform repetitive operations on similar matrices.

IV. EXPERIMENTAL EVALUATION

In evaluating the accuracy of the model for singular value predictions, we split the original data into training and testing. In looking for possible sets to train and test our regression model on, it is important that we use real data instead of artificially constructed data. This is since we get matrices of varying ranks but still with enough significant patterns that our neural network can learn and further predict on. With the CIFAR-10 dataset [15], we randomly selected $50,000$ images for training and $10,000$ for testing. On the Spambase [16] dataset, we randomly chose $104,858$ and $18,505$ (about

15% of the whole dataset) matrices for training and testing, respectively.

A. Metrics

Below, we evaluate our proposed model in a number of different dimensions: (1) Comparison between the real and predicted singular values, (2) Matrix reconstruction error, and (3) Visual comparison between the true and reconstructed images.

Comparison of singular values: Our first method in validating the effectiveness of our proposed model is to directly compare the real and predicted singular values. In the testing phase, we feed in the flattened matrices to the trained neural network receiving an array of 32 predicted singular values for every test matrix. For a numerical comparison, we used normalized mean squared error (NMSE) and mean absolute error (MAE) of all the test matrices, which, for each matrix, are defined in Eq. 1 where $\{\sigma_i\}_{i=1}^N$ and $\{\hat{\sigma}_i\}_{i=1}^N$ are the real and estimated singular values with N representing the total number of singular values.

$$\text{NMSE} = \frac{\sum_{i=1}^N (\sigma_i - \hat{\sigma}_i)^2}{\sum_{i=1}^N \sigma_i^2}, \qquad \text{MAE} = \frac{\sum_{i=1}^N |\sigma_i - \hat{\sigma}_i|}{N} \tag{1}$$

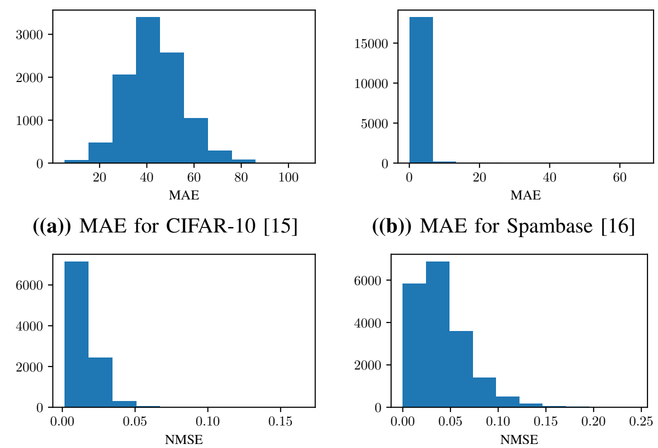
In Table I, the performance of our proposed method in estimating the singular values of matrices from CIFAR-10 [15] and Spambase [16] datasets are reported. Specifically, we present the average and standard deviation of NMSE and MAE among all the test samples, which shows that our neural networks can be used to accurately and consistently predict singular values. In order to fully present our prediction performance, we also show the error histograms in Figure 2, which shows fairly low and consistent errors giving insight on the accuracy of our model despite having a moderate amount of data to train on. In the future, these values could be improved on with a more advanced deep learning model and more training data.

Furthermore, we conduct a direct comparison between the true and predicted values of some randomly chosen data from the CIFAR-10 and Spambase datasets. In Figure 3, we plot the predicted and true singular values in a logarithmic scale versus their indies in the top 4 panel, the predicted and true singular values in a linear scale versus their indies in the middle 4 panel, and the gray-scaled images in the bottom 4 panel followed by their corresponding MAE of the predicted singular values, where each column represents the results of a specific image sample. Clearly, our predicted singular values are very close to the real ones, and the significant ones are almost identical the real values indicating that the proposed method is a good tool to predict the rank of a matrix. Similar conclusions can be drawn from the testing results of Spambase data in Figure 4.

Matrix reconstruction: Another method in evaluating our algorithm is by reconstructing matrices using the predicted singular values combined with the true singular vectors. We performed this specifically on the CIFAR-10 datasets as it allows us to visualize the reconstructed images as well. Specif-

Dataset	NMSE	MAE
CIFAR-10 [15]	$1.59\% \pm 0.89\%$	43.36 ± 11.95
Spambase [16]	$4.04\% \pm 2.89\%$	1.00 ± 2.14

TABLE I: The NMSE and MAE of the predicted singular values; $a \pm b$ where a and b are mean and standard deviation.



((a)) MAE for CIFAR-10 [15] ((b)) MAE for Spambase [16]

((c)) NMSE for CIFAR-10. ((d)) NMSE for Spambase.

Fig. 2: Error histograms of our proposed method for CIFAR-10 and Spambase. The plot in Fig 2(a) shows a fairly normal distribution of absolute error indicating consistency. Although the plot in Figure 2(b) is not a normal distribution like 2(a), the MAE still remains consistent. The plot in Figure 2(c) shows low NMSE where most errors are in the $0 - 5\%$ range. The plot in Figure 2(d) shows the errors ranging mainly from 0-10%, which combined with the results in Figure 2(c) demonstrate that our method generalizes to different types of datasets.

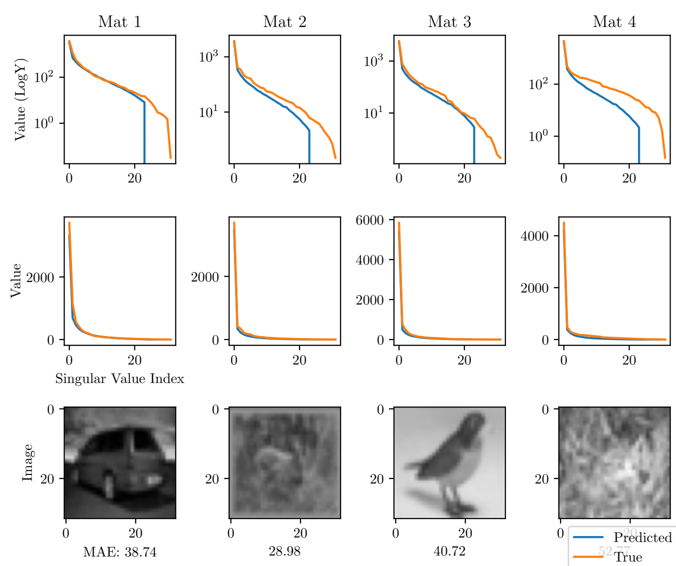


Fig. 3: Direct comparison of the predicted and true singular values of a CIFAR-10; see the logarithmic singular values and the singular values in the first and second rows, respectively. We can see that our method generally approximates the singular values well, but tends to falter by predicting 0 for smaller values.

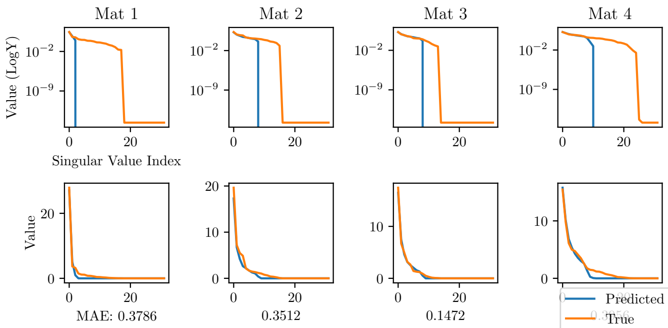


Fig. 4: Direct comparison of the predicted and true singular values of the Spambase dataset; see the logarithmic singular values and the singular values in the first and second rows, respectively. We can see that our model generally predicts the singular values accurately for the first few significant values but tends to predict 0 for more numbers than the real SVD.

ically, we plot: 1) the reconstructed images using SVD in the top 4 panels in Figure 5, where the number of singular values is set as $k = 1$, 2, 5 and 32; and 2) the reconstructed images using the same left and right singular vectors as 1) as well as the top k singular values obtained from our proposed method which are shown in the bottom 4 panels of Figure 5. From the results in Figure 5, we can see that our predicted singular values are accurate even in low rank image reconstruction. This comes to show a feasible application such as the SVD-LIGHT where we can utilize a much smaller amount of data with the neural network's approximations to yield similar results as the real SVD. It's worth to mention that for the randomly chosen image in Figure 5, the estimated rank is 6 as the image reconstruction error begins to level off at around 6 rows and columns out of the original 32×32 matrix, which matches our observation that the image reconstruction at $k = 5$ is very close to that using the full rank, i.e., $k = 32$. The combination of SVD-LIGHT method and our proposed singular value estimation method act as a case study for a possible application of data reconstruction in a faster setting.

Visual comparison: In Figure 6, we compared the reconstructed images (three image examples) using the real, random, and network prediction singular values from our method as well as the true singular vectors; see the results in the first three columns of Figure 6, and the SVD-LIGHT 1 method; see the results in the last column of Figure 6. We also used a comparison with an image-reconstruction using random singular values in order to determine that our predicted sigma is not simply outputting random numbers and giving characteristic data. Our results show a drastic similarity when using our model's predictions and the real singular values for matrix reconstruction. There are slight differences such as the darkness or contrast, though the details and overall image are maintained when using our neural network. In comparing the random singular value and SVD-LIGHT reconstructions, there are worse results and unintelligible details.

Overall, there is a very minimal error when using a neural network to predict singular values. Especially with medium-

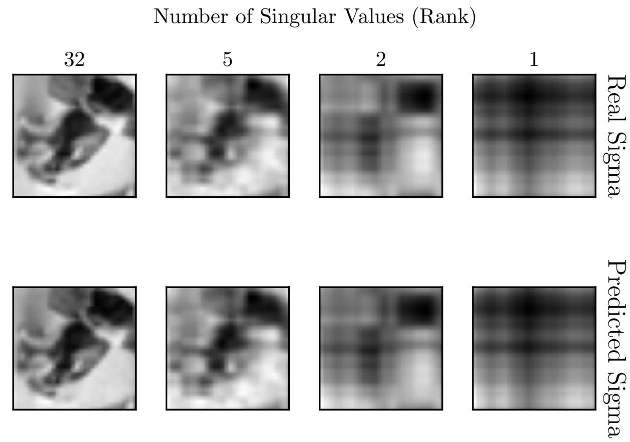


Fig. 5: Reconstruction of a CIFAR-10 [15] image using various number of singular values (a.k.a., ranks where $k = 1$, 2, 5 and 32) from the real and predicted singular values (a.k.a., Sigma in the figure), and the true singular vectors. The neural net's singular values serve as an accurate alternative especially at low ranks of 5 and up.

sized datasets under 100,000 instances, these results are indicative that the use of neural networks has the possibility of being used as a practical method in reducing run-time and handling large input datasets over the SVD.

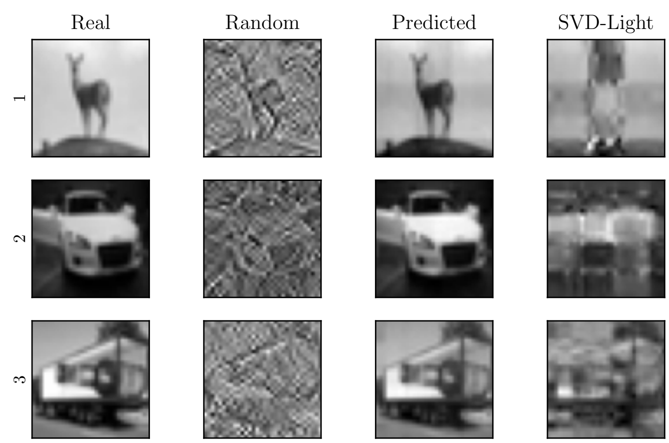


Fig. 6: Three image reconstruction examples from the CIFAR-10 [15] dataset. The first three columns are the reconstructed images using using real, random, and model predicted singular values and the true singular vectors. The last column shows the results of the SVD-LIGHT method.

B. Discussion

Our experiments cover the prediction of the singular values of matrices but do not cover the full decomposition for the singular vectors. In the future, this same process can be applied to find these matrices to make a neural network that is completely independent from the SVD after training. Our datasets were also not very large to the point where our models can be applied to all different types of data. When training, we specialized our network to correctly predict the singular values of matrices with similar properties. This can be improved by providing much more data to train so that the application of that model is broader. The paper also focuses more on the feasibility of using a neural network to find singular values and not exactly the time it takes for the training and further predictions. Though, further work can focus on lowering the time and resource usage for large dataset applications of the SVD by creating more optimized models.

V. Related Work

On the note of faster SVD methods, [18] proposes a method of using a mix of Monte Carlo (repeated random sampling) and empirical sampling to use a subset of a large-scale matrix to approximate the SVD within an error bound. Similar to our SVD-Light their method involves taking parts of a matrix but further decreases error by adjusting the variance of the stratified data to meet the target accuracy, finding a more efficient method of sampling for the SVD. This gives an idea on how our neural network can practically decrease run-time but also yield accurate results by decreasing the amount of data used for the decomposition for left and right singular vectors while also using the model to give accurate singular values based off the entire original matrix.

[19] and [20] propose a similar method where they use a subset of the columns of matrices but repeat this with using fixed and adaptive sampling schemes to decrease the error bounds when generating low-rank matrix approximations. They add on to this work by applying the SVD on a submatrix of a large matrix using a randomized low-rank approximation algorithm to effectively maintain the accuracy of a large SVD with the time complexity of a small SVD. Like SVD-Light and [18], a possible addition that our work can make to this is using a neural network to calculate accurate singular values based off the entire matrix while using their method to perform the rest of the SVD. If a model can be used to predict the entire SVD decomposition, then the complexity of using SVD can be eliminated altogether with the [19].

VI. Conclusion & Future Work

While there is still a lot more work to be done in the area, this work shows the feasibility of using a supervised neural network to accurately predict the singular values of a given matrix and the future application of a faster and lighter alternative to the singular value decomposition. Throughout our process, we explored the application of neural regression models on visual data sets such as the CIFAR-10 [15] and numerical sets such as UCI's Spambase [16].

For evaluation, we utilized a mix of visual comparison in using the CIFAR-10 data [15] and direct comparisons for the Mean Absolute Error and Normalized Mean Squared Error between our predicted and SVD singular values. As a result of these tests, we found significant results giving us singular values that were very close to what the SVD could compute. When comparing our singular values to randomized data or our SVD-Light1, the neural network outperformed these methods drastically, showing the feasibility of using trained models instead of the typical SVD algorithm, with errors between 1-5% and under 100,000 instances to train on.

Thus far, we have presented a proof-of-concept framework, demonstrating the viability and plausibility of our original goal. In the future, we envision that these findings will make way for a much faster alternative and lighter alternative to the SVD for applications in large data where the time complexity of the SVD becomes too significant. We defer this exploration for future work, where major interesting challenges will include ways to best represent the input data, and designing the most appropriate architecture which may leverage the structure of the problem in order to learn more efficiently.

VII. Acknowledgements

Research was supported by the National Science Foundation under CAREER grant no. IIS 2046086 and CREST Center for Multidisciplinary Research Excellence in Cyber-Physical Infrastructure Systems (MECIS) grant no. 2112650, and UCR Regents Faculty Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

References

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[2] F. Gao and A. B. Gershman, "A generalized esprit approach to direction-of-arrival estimation," *IEEE signal processing letters*, vol. 12, no. 3, pp. 254–257, 2005.

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, Sep. 1990.

[4] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos, "Eigenspokes: Surprising patterns and scalable community chipping in large graphs," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2010, pp. 435–448.

[5] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[6] B. T. Nebgen, R. Vangara, M. A. Hombrados-Herrera, S. Kuksova, and B. S. Alexandrov, "A neural network for determination of latent dimensionality in non-negative matrix factorization," *Machine Learning: Science and Technology*, vol. 2, no. 2, p. 025012, 2021.

[7] H. Prasantha, H. Shashidhara, and K. Balasubramanya Murthy, "Image compression using svd," in *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, vol. 3, 2007, pp. 143–145.

[8] S. K. Singh and S. Kumar, "A framework to design novel svd based color image compression," in *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, 2009, pp. 235–240.

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[10] M. W. Berry, "Large-scale sparse singular value computations," *The International Journal of Supercomputing Applications*, vol. 6, no. 1, pp. 13–49, 1992.

[11] D. Yang, Z. Ma, and A. Buja, "A sparse svd method for high-dimensional data," 2011. [Online]. Available: https://arxiv.org/abs/1112.2433

[12] I. Hegedűs, M. Jelasity, L. Kocsis, and A. A. Benczúr, "Fully distributed robust singular value decomposition," in *14-th IEEE International Conference on Peer-to-Peer Computing*, 2014, pp. 1–9.

[13] M. A. Iwen and B. Ong, "A distributed and incremental svd algorithm for agglomerative data analysis on large networks," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 4, pp. 1699–1718, 2016.

[14] H. Mehta, S. Rendle, W. Krichene, and L. Zhang, "Alx: Large scale matrix factorization on tpus," 2021. [Online]. Available: https://arxiv.org/abs/2112.02194

[15] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[16] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase," UCI Machine Learning Repository, 1999.
[17] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[18] M. Holmes, A. Gray, and C. Isbell, "Fast svd for large-scale matrices," 01 2007.
[19] M. Li, W. Bi, J. T. Kwok, and B.-L. Lu, "Large-scale nyström kernel matrix approximation using randomized svd," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 152–164, 2015.
[20] S. Kumar, M. Mohri, and A. Talwalkar, "Sampling techniques for the nystrom method," in *Artificial intelligence and statistics*. PMLR, 2009, pp. 304–311.