Towards better visual exploration of large high-dimensional data

1 Introduction

1.1 Motivation

In the age of data science, interactive visualization of large high-dimensional data is an essential tool for efficient data mining and instant knowledge extraction. It allows for both the insight into data structure and its interactive exploration by direct manipulation on the whole or a fragment of a dataset. This way, it is possible to observe the topological properties of classes and their mutual locations, as well as remove irrelevant data samples and identify the outliers. The multiscale structure can be explored visually by changing data embedding strategies and visualization modes (e.g., the type of the loss function), and zooming-in and out selected fragments of 2(3)D data mappings. When we consider multidimensional feature vectors as an input representation, naive searching for the correspondence between higher-level features in the black-box classifier and the raw input vectors very often does not provide any explanations. It can be caused by the features of the input data, that are not directly interpretable, such as paragraph vector representation. This is where data visualization may facilitate interpretability in otherwise difficult problems. However, one major obstacle here is the difficulty of projecting high-dimensional data into two or three dimensions that preserves all important (local and global) structural properties of the source ND data in 2(3)D space. Visualization is extremely important in the context of other types of embeddings, i.e., the search for optimal vector representation of data (in the case of unstructured data such as images, text, graphs) and looking for the best procedure of manifold learning.

1.2 Problem definition

Data embedding (DE) (called also Vector Embedding (VE) we use this acronym for multidimensional data visualization) is defined as a transformation $\mathbf{B} : Y \rightarrow X$ of N-dimensional (ND) dataset $\Re^N \ni Y = \{y_i\}_{i=1,...M}$ into its n-dimensional (nD) representation $\Re^n \ni X = \{x_i\}_{i=1,...M}$, where $N >> n$ and M is the number of ND feature vectors y_i and corresponding nD embeddings x_i. The mapping $\mathbf{B}$ can be perceived as a lossy compression of data. It is performed by minimizing a loss function $E(\|Y - X\|)$, where $\|.\|$ is a measure of topological dissimilarity between Y and X. Due to the high complexity of the low-dimensional manifold, immersed in the ND feature space and occupied by data samples Y, perfect embedding of Y in the nD space is possible only for trivial cases. Furthermore, for many of the state-of-the-art (SOTA) embeddings [2, 34, 54, 57]:

1. The time complexity of the DE procedure is dominated by computing $O(M^2)$ distance tables for both the source data and their low-dimensional target representations.
2. The computational efficiency of the DE process vastly depends on the loss function and optimization procedure applied for its minimization.
3. Calculating gradient of the loss function is $O(N \cdot M)$ complex, but with large proportionality coefficient.

1.3 Various types of data embedding

Searching for data representations in Euclidean vector space. This group of embedding algorithms contains methods, which create data representation of unstructured data (e.g., text, graphs,

pictures) on the basis of semantics (word2vec [35], doc2vec [25], node2vec [17], graph2vec [38] etc.) so called, paragraph vectors. The method being the precursor of all the others is word2vec. It takes as its input a large corpus of text and produces a vector space, typically a few hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space. All the other methods incorporate analogous idea for different use cases (e.g. embedding whole documents, graphs, pictures, etc.), which can then be used for various downstream machine learning tasks.

Creating a compressed representation of high-dimensional data. The current SOTA methods for creating shorter data representation are: Autoencoders [51], Variational Autoencoders [46] (having additional generative properties) and methods of manifold learning (Isomap [49], Laplacian eigenmaps [5], UMAP [34] etc.), which are trying to unfold a low-dimensional geometrical structure of the low dimensional manifold flooded in the feature space.

Autoencoders (left side of Fig. 1) is an unsupervised artificial neural network that learns how to efficiently compress and encode data. When this process is completed, it learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible. The network architecture for autoencoders can vary between a simple feedforward network, LSTM [19] or Convolutional Neural Network [1] depending on the use case.

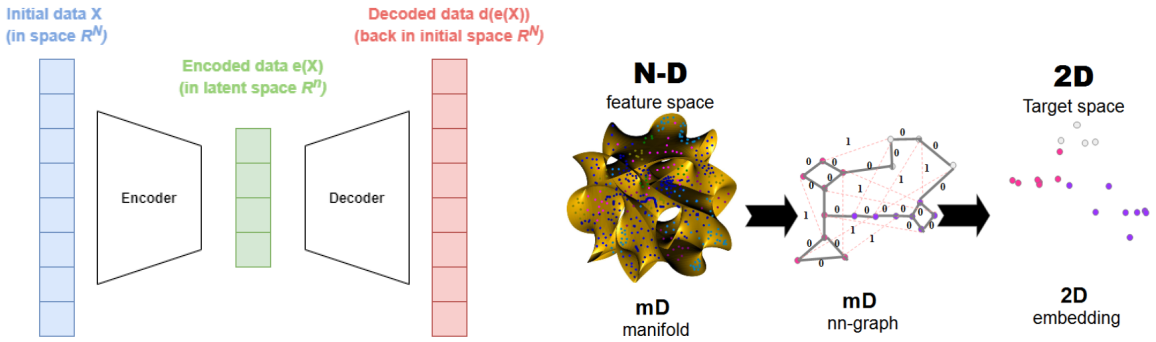


Figure 1: Left: high-level autoencoder architecture, where data dimensionality $N>>n$. Figure taken from [50]. Right: the idea of manifold learning (in the context of visualization) by means of the nearest neighbor graph [11].

Variational Autoencoder (VAE) is the autoencoder whose encodings distribution are regularised during the training in order to ensure that its latent space has smooth properties (the space is regular) allowing us to generate some new data. Moreover, the term "variational" comes from the close relation there is between the regularisation and the variational inference method in statistics [7].

Manifold learning (MaL) [16] (right side of Fig. 1) is an approach to nonlinear dimensionality reduction (feature extraction). Those algorithms are based on the idea that the intrinsic dimensionality of many data sets is small and data occupies a low-dimensional manifold, due to complex and nonlinear interdependency between features [4]. Therefore, each pattern can be potentially described by much smaller number of features n ($n<<N$). The principal difference between autoencoders and MaL feature engineering (FE) is in totally different approach to the loss function definition. For the autoencoders we require that the source data can be reconstructed from the target ones. In the case of MaL based FE, the mapping to low-dimensional target space is irreversible, and we intend to reconstruct the basic dataset structure, i.e., some local and global data properties such as the neighborhood and data separation will be preserved in the target space. Manifold learning can be considered as a generalization of the multidimensional scaling (MDS) based group of algorithms and other frameworks such as LLE [26], Isomap [49], spectral embeddings [40], t-SNE [32] and their clones (e.g., [18, 28, 44]). Though supervised variants of data embedding exist [27], the typical manifold learning problem is unsupervised: it learns the high-dimensional structure of data from data itself, without any use of predetermined classifications. The general steps for creating these types of algorithms are as follows:

1. Closeness, neighborhood, clustering: Find groups of similar points. Given input data $X = \{x_1, \ldots, x_n\}$, build a function $f : X \rightarrow \{1, \ldots, i\}$, where i is an index of a cluster and $i \in \mathcal{N}$. Two close points should be in the same cluster (have the same value of i).
2. Separation: Find distant points. Distant points cannot be in a single cluster.
3. Dimensionality Reduction: Project points into a lower dimensional space while preserving the structure of original data (closeness and separation should be preserved). Given $X = \{x_1, \ldots, x_n\} \in R^N$, build a function $f : R^N \rightarrow R^n$, where $N >> n$.
4. Given labeled and unlabeled points, build a labeling function for $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, $f : X \rightarrow Y$. Two close points should have the same label.

Visualization Main target of visualization (see Fig.1 (right)) is to create a projection of ND space into visually perceived $2(3)D$ Euclidean space. Visual exploration of large multidimensional datasets is a special case of the manifold learning (MaL) and feature extraction problem. However, as we told in the Motivation section its role is quite different. By visualizing data we do not focus on decreasing (condensing) the number of features to a realistic number representing approximate dimension of the manifold. Instead, we produce mapping in extremely low dimensional space (2(3)D) which should be perceived by human. So, unlike in the classical MaL algorithms the data embedding (DE) algorithms should focus on visual aspects, allowing human for instant learning of data structure via interactive visual exploration of its multidimensional topology.

Summarizing, interactive visualization should allow for: 1) instant verification of a number of hypotheses, 2) precise matching of data mining tools to the properties of data investigated, 3) adapting optimal parameters to machine learning algorithms, and 4) selecting the best data representation. Herein, we focus on application of data embedding (DE) methods in the interactive visualization of large ($M \sim 10^5$-10^6) and high-dimensional $N \sim 10^{2+}$ data.

2 Research project objectives

Interactive visual exploration of big multidimensional data still needs more efficient $ND \rightarrow 2D$ DE algorithms and integration of the most powerful DE algorithms on a single computational platform. The application of DE and interactive visualization for scrutinizing ANNs in the context of their interpretability, is also an interesting, not yet fully explored, computational challenge. Therefore, our research program focuses on these three main research objectives, i.e.,:

1. Developing, improving and accelerating IVHD [11] (interactive visualization of high-dimensional data) method designed in the scope of our previous NCN grant[1].
2. Creating an integrated computational platform (library and GUI), which would allow for visualization and interactive exploration of large and high-dimensional data by integration of various DE algorithms, optimization methods and loss functions.
3. Adopting the platform as a tool for interpretation of decisions of neural networks (particularly deep neural networks) by visualizing the dynamics of data representations and hidden activity of NN during training (e.g. in various NN layers), and looking for activation of neurons to define their responsibility for NN decision making.

One of the main drawbacks of modern, and very accurate in terms of reconstruction local structure of data (such as t-SNE [32]), multidimensional DE algorithms is that they are still too computationally demanding in terms of computational speed and storage, to use them in interactive way for visual exploration of truly big data. This is due to the character of the most data sets, which requires using nonlinear dimensionality reduction algorithms, which in general have much higher computational complexity than linear ones and generate better low-dimensional representations. As presented in [37], both the computational and storage complexities can be radically decreased

[1] 2013/09/B/ST6/01549 : Interactive Visual Text Analytics (IVTA): Development of novel user-driven text mining and visualization methods for large text corpora exploration (2013-2016).

by drastic simplifications of t-SNE assumptions. On the other hand, these simplifications towards visualization of larger and larger datasets in a shorter time, can be made at the expense of their accuracy. Especially, concerning preservation of the local data structure. Therefore, for interactive visualization the integration of many DE algorithms in the scope of a single computational platform is so important. Having, the general view on the data structure we can focus on its fragments by using more "locally-oriented" algorithms.

2.1 Developing, improving and accelerating IVHD method

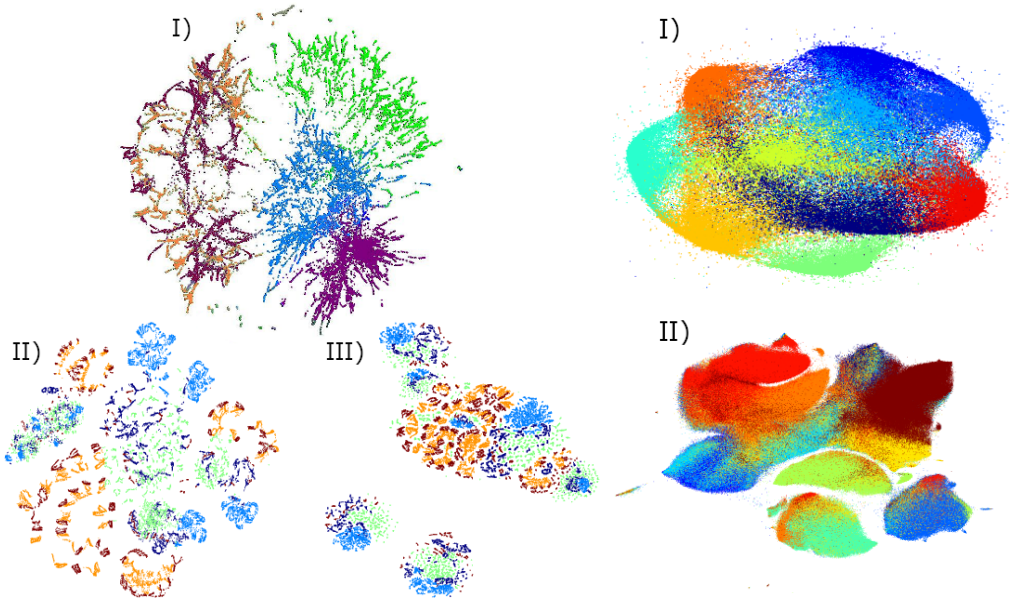
As we have emphasized above, the state-of-the-art (SOTA) methods used for N-dimensional data embedding to the visually perceptible Euclidean space (2D or 3D), such as those based on t-SNE concept, are too demanding computationally for interactive analytics of data consisting of $M \sim 10^{6+}$ feature vectors and dimensions $N \sim 10^{2+}$. The method developed by our research team - **IVHD** (**I**nteractive **V**isualization of **H**igh-dimensional **D**ata) [11, 12], outperforms the modern data-embedding algorithms in both computational and memory loads, while preserving in 2D (3D) main topological properties of original ND data such as class separation and their mutual proximity. As shown in our earlier publications [11, 12] and particularly the most recent one [37] (code repository on github: [41]), embedding of high-dimensional ND data to 2D (3D) can be reduced to the problem of the kNN-graph visualization (GV), where nodes represent the feature vectors. However, unlike the other SOTA DE algorithms, which also employ this trick, IVHD method use very small value of k and information about distances between samples in original ND space can be discarded after kNN-graph construction. These radical assumptions result in the linear time&memory data embedding with very small proportionality coefficient, much smaller than those in the linear time&memory SOTA DE methods.

IVHD For truly high-dimensional data ($N \gtrapprox 30$), i.e., when the influence of "curse of dimensionality" on the ND space topology become evident (the random distances between vectors are approximately the same), the floating point dissimilarities used for kNN-graph construction can be discarded. Instead, the integer indices to only a few k nearest neighbors have to be kept in the computer memory. We have showed [11] that for each ND feature vector we :

1. Store only indices of k nearest neighbors, that provide rigidness and coherency of created kNN-graph ($k = 2$ in the most cases is sufficient).
2. Select very few (often just one) random neighbors *rn* (similar to *negative sampling* procedure) during calculations.
3. Define binary distances (0 and 1) to the nearest and random neighbors, respectively.
4. Use MDS and particle based optimizer [12] for visualization, having in mind that the number of distances used in simulation is drastically smaller (proportional to M) than in both classical and most recent [3, 42] MDS realizations.

As shown in [37], we successfully and radically accelerated the calculation process by implementing IVHD method in GPU/CUDA environment. We have been optimizing it even further for CPU (SSE, AVX implementation). As shown in Table 1 of [37] the GPU version of our code is one order of magnitude faster than the baseline algorithms (including kNN-graph generation) and up to 30 times faster in pure embedding (without the procedure of kNN-graph generation, which is the same for all the methods) of the largest ($M = 1.4 \cdot 10^6$) YAHOO dataset (comparing to AtSNE-CUDA [15] the baseline and the state-of-the-art GPU/CUDA implementation of t-SNE concept). In general, the quality of embedding is a bit worse. Especially, for smaller M in preserving the local structure of data. However, as shown in [13], the quality of the local neighborhood for target representation of data is still high for IVHD.

As we demonstrate in Fig. 2a, for SmallNORB dataset IVHD-CUDA was able to visualize clearly separable three big clusters with fine-grained data structure. BH-SNE-CUDA and AtSNE-CUDA generated embeddings, that are more fragmented and worse in terms of quality. The changes of the parameter values of the baseline algorithms (e.g., perplexity) did not improve the visualization quality. For Yahoo dataset (Fig. 2b) IVHD-CUDA generates more fuzzy output than AtSNE-CUDA, but the samples from the same class are closer together than those generated by the baseline



(a) SmallNORB (M=48600, N=2048) (b) Yahoo (M=1.4·10^6, N=100)

Figure 2: Visualizations of datasets using: I) IVHD-CUDA, II) AtSNE-CUDA , III) BH-SNE-CUDA. BH-SNE-CUDA did not generate visualization for bigger datasets.

method. Meanwhile, AtSNE-CUDA was able to separate clearly visible but fragmented clusters. What is worth to mention - timings of IVHD-CUDA method was considerably shorter for both visualizations [37]. In IVHD-CUDA we also implemented a few SOTA optimization schemes such as: Nesterov, Adagrad, Adadelta, RMSprop, NAG, Adam and force-directed (F-D) method [12] to calculate minimum of the IVHD loss function. As shown in Table 1 [37], the force-directed (F-D) approach and synchronized Nestorov scheme appear to be the best in terms of accuracy (i.e., a better minimum of the loss function reached).

To compare data separability and class purity, we define the following simple metrics:

$$cf_{nn} = \frac{\sum_{i=1}^{M} nn(i)}{nn \cdot M} \quad \text{and} \quad cf = \frac{\sum_{nn=1}^{nn_{\max}} cf_{nn}}{nn_{\max}}, \tag{1}$$

where $nn(i)$ is the number of the nearest neighbors (k) of x_i in X space, which belong to the same class as y_i. To reflect a wide range of embedding properties, we use $nn_{\max}$=100.

Preliminary research results Table 1 presents the results obtained by the IVHD-CUDA in comparison with the SOTA GPU implementations of DE algorithms [37]. It is clear, that our method is the fastest one for the baseline datasets. In general, the local structure of the *source* data is better reconstructed by the two baseline SNE algorithms. Nevertheless, IVHD better preserves the class structure (it is not so fragmented) and its relative locations. For fine-grained structures of classes (such as in the SmallNORB dataset), IVHD outperforms its competitors in both the efficiency and - slightly - in the cf accuracy. Moreover, unlike AtSNE-CUDA and BH-SNE-CUDA, IVHD is able to visualize the separated and not fragmented classes.

Despite radical simplifications, IVHD method still properly reconstructs the main structural properties of large ND datasets at the cost of rather minor and, incomparable to the scale of these radical approximations deterioration, embedding quality. In our opinion, it is the most important result obtained in [37], which shows how robust is the "backbone" of high-dimensional data represented by its very sparse (k is small) kNN-graph. It is interesting, how robustness of this ”backbone” is correlated to the complexity of a low-dimensional manifold occupied by data samples and embedded in a very high-dimensional feature space. Meanwhile, the algorithms based on t-SNE

Table 1: The values of cf_{nn} accuracy for various datasets and GPU/CUDA implemented embedding methods. The timings show: the overall embedding time (*time*), *k*NN-graph generation time ($time_{gg}$), net embedding time ($time_{emb}$). The results are the averages over 10 simulations.

Dataset	Algorithm	*time* [s]	$time_{gg}$ [s]	$time_{emb}$ [s]	cf_2	cf_{10}	cf_{100}
MNIST	BH-SNE-CUDA	32.588	**5.813**	26.775	0.94	0.938	0.933
	AtSNE-CUDA	15.980		10.167	0.944	**0.943**	**0.938**
	IVHD-CUDA	**7.326**		**1.261**	**0.946**	0.936	0.924
FMNIST	BH-SNE-CUDA	32.913	**6.734**	26.179	0.757	0.755	**0.738**
	AtSNE-CUDA	17.453		10.719	0.76	**0.757**	0.737
	IVHD-CUDA	**8.177**		**1.443**	**0.767**	0.726	0.670
SmallNORB	BH-SNE-CUDA	38.673	**15.517**	23.161	0.944	0.919	0.745
	AtSNE-CUDA	20.521		5.009	**0.97**	**0.94**	0.73
	IVHD-CUDA	**16.151**		**0.634**	0.936	0.921	**0.828**
RCV-Reuters	BH-SNE-CUDA	-	**45.302**	-	-	-	-
	AtSNE-CUDA	220.39		175.088	0.82	0.82	**0.818**
	IVHD-CUDA	**60.72**		**15.418**	**0.835**	**0.828**	0.803
YAHOO	BH-SNE-CUDA	-	**52.930**	-	-	-	-
	AtSNE-CUDA	628.63		575.7	**0.686**	**0.686**	**0.686**
	IVHD-CUDA	**70.12**		**18.930**	0.668	0.662	0.653

still assume that Euclidean distances are responsible for the structure of data. But this is not true at all for very complex manifolds resulting from strong feature interdependence. As a result, they too often produce very fragmented visualizations. In our opinion, just the sparse *k*NN-graph (small *k*) is the most appropriate structure, which is able to approximate such a complex manifolds [34]. We believe that our method is very helpful for interactive visualization of truly big and complex data, where low storage and high computational speed of DE algorithm are the crucial issues. For those reasons, we consider further development of IVHD method to have a positive impact in the community related to machine learning, big data embedding and visualization. Therefore, we see at least a few ways for further improvement:

1. We can extract more information about *k*NN graph structure and pass it to visualization algorithm. However, we have to assume that this procedure will not increase considerably the simulation time and make the embedding process more complex. To this end we consider:
   - utilizing reverse *k*NN mechanism [55] (as shown in our preliminary results it improves data clustering due to additional repelling forces for reverse neighbors);
   - fast pre-clustering of data, e.g., using shared nearest neighbor (SNN) clustering concept [29], what allows for using different weights for *k*NN-graph nodes and edges depending on the local first and secondary nearest neighborhood;
2. We will check the possibility of hybridization of variational autoencoders (VAE) with data embedding procedures to increase the quality of data visualization employing VAE generative power and making the target space more regular,
3. The most of modern multidimensional data visualization algorithms are unsupervised. However, in many cases we have additional knowledge about data, which can help in its better visualization. We will develop the algorithms and practices for partly supervised visualization of multidimensional data,

2.2 Platform for visualization of high-dimensional data

Platform development process consists of two key components: 1) creating a library and 2) creating a web service, that uses the library. Prototypes of platform architecture are being developed. The library itself will be built upon described above IVHD method. We plan to integrate also the most successful data embedding methods (e.g., LargeVis [54], bh-SNE [31], UMAP [34], TriMap [2],

ShapeVis [24]) to enable the change of visualization view from more global to more local, where more demanding and more accurate DE algorithms can be used. Thus, the components of the library will be closely interrelated, but it will be possible to use them independently.

The integration of SOTA DE methods in the common framework would allow to create a stable environment for: 1) testing and evaluating embedding and optimization methods, 2) adjusting parameteres used by different methods, 3) formulate and instantly verify number of hypotheses and 4) integrate the methods in one visualization framework allowing for multi-scale analysis of data, employing more accurate methods for finer scales.

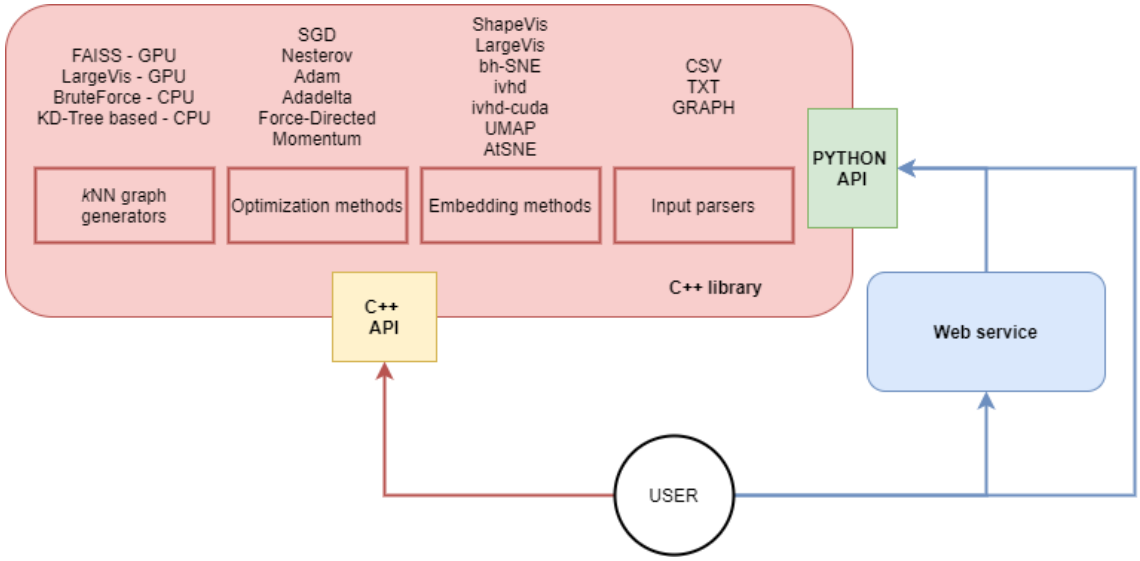


Figure 3: The high-level scheme of the interactive visualization platform. The user can access the possibilities offered by the library in three ways: 1) by using C++ API directly, 2) by using Python API directly and 3) by the web service implemented in Python. By using public interfaces it is possible to create a new implementations, that will be added to the library (e.g., when scientists would like to test their new method).

The next step of the platform development would be to devise a cloud infrastructure to perform server-side calculations and analysis. It would allow users to analyze really large datasets (which cannot be stored in the local memory). To do that, the Python API is being developed (this step also allows to use the IVHD library with most popular Python frameworks and utilize all capabilities presented by its data processing modules). As addition to library itself - an Editor tool is being created in Qt environment. It allows to visualize and manipulate datasets locally by using graphical user interface (GUI).

Currently, it is possible to specify kNN graph generation algorithm [21, 23] and a few current SOTA optimization methods has been implemented, i.e., Nesterov [39], Adagrad [10], Adadelta [58], Adam [22], Momentum [9]. Library also implements the force-directed approach (originally used in [12]).

As presented in Fig. 3 the library (red rectangle) consists of four main blocks, which are responsible for: 1) fast generating kNN graph, 2) specifying optimization method used by the DE algorithm, 3) specifying DE algorithm and 4) parsing the input from different types of file (CSV is the most common one for storing datasets in human-readable format). Depending on the use case it is possible to use library directly through implemented C++/Python application programming interface (API) or through web service. It is worth to mention that IVHD algorithms currently implemented in the library are capable of utilizing both CPU and GPU-CUDA infrastructure [37, 41]. Furthermore, FAISS library [21] is embedded into IVHD library for fast generation of kNN-graph.

2.3 Adopting the platform as a tool for interpretation of decisions carried out by multi-layer neural networks

Let D be a dataset, which is composed of pairs (x, y), where $x \in R^m$ is an observation, and $y \in \{0, 1\}^d$ is a target class assignment. We can consider various kinds of ANNs. Firsly, we will focus on multilayer perceptrons (MLPs). Such a network represents a parameterized function $f : R^m \rightarrow (0, 1)^d$, which usually attempts to generalize class assignments from examples in D. The root of the problem is the fact that Neural Networks are vastly overparametrized [20]. Reducing this complexity may go a long way towards more interpretable models. Sometimes, the neural network models tend to have orders of magnitude more parameters than the number of examples used to train them. This overparametrization have profound impact on their interpretability. Our proposal address the following two tasks:

1. Exploring the relationships between alternative representations of observations learned.
2. Exploring the relationships between artificial neurons.

Simultaneously, we would like to do that for paragraph vector embeddings, what makes the interpretability even more complex. The features in paragraph vectors have no meaning. Using other words, finding the neurons responsible for NN decisions, does not explain what they really mean. To answer this question we will try to use visualization for examining the activity of hidden layers in skip-gram or c-bow networks used for generating data representation [36]. In order to do that, we need very fast and easy-to-use method of data visualization (presented in the previous section).

Preliminary research results Using the approach proposed in [47] as a starting point for visual interpretation methodology, we briefly show how this kind of hidden layer visualizations can provide valuable feedback for model interpretation. To this end we have trained multilayer perceptron models on a popular benchmark: newsgroup posts (M=14580, N=2048). We construct visualizations using our fast embedding algorithm [12]. Hidden representations for newsgroup posts dataset exhibit not very good clustering structure (Fig. 4, left and center). However, the same model compressed with a standard algorithm provided by a popular deep learning framework display vastly more structured hidden representations (Fig. 4, right). This result suggest that compression and complexity reduction may go a long way towards improved interpretability of neural models. Fig. 5 presents embeddings of hidden representations in the multilayer perceptron model for 6-NG test subset (6 class out of 20 from 20NG dataset) along with their discriminative power (defined as the accuracy of a simple k-nearest neighbor classifier trained on hidden representations). We can observe, which neurons are responsible for assigning objects to a given class. By performing such analysis it is possible to extract crucial features and remove superfluous neurons.

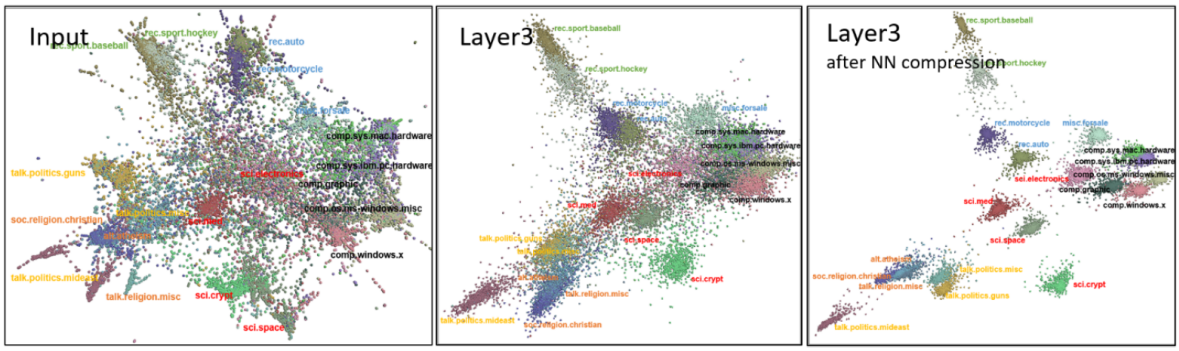


Figure 4: Two dimensional embeddings of newsgroup posts representations in a deep neural network. Left and center: embeddings for a fully-connected multilayer perceptron model. Right: embeddings for a compressed multilayer perceptron model.

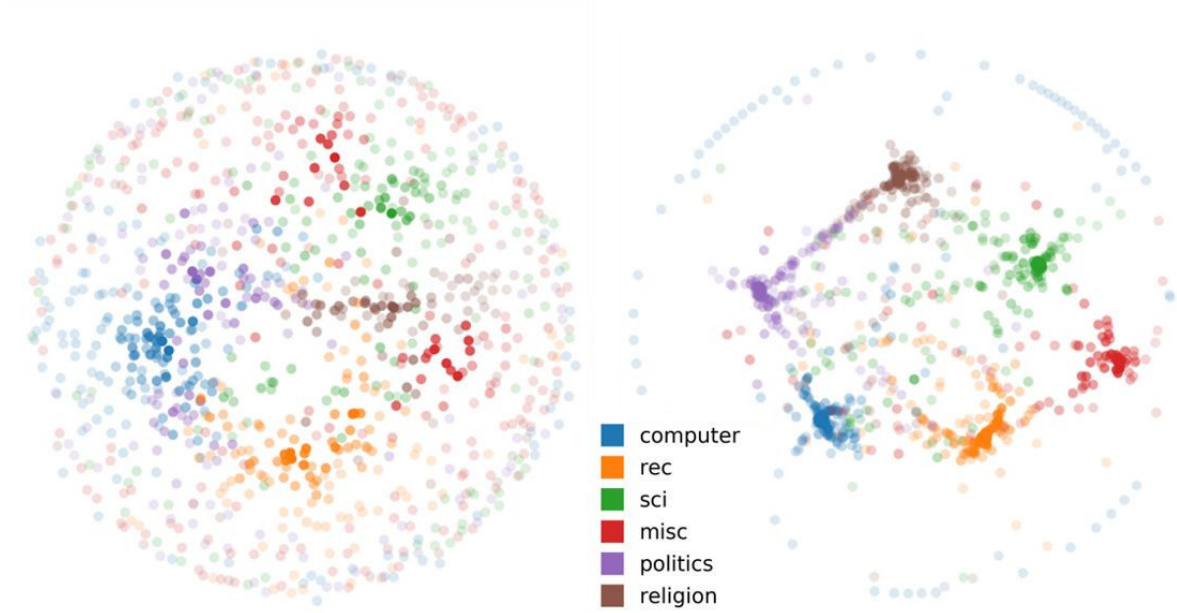


Figure 5: Discriminative neuron activation maps of the first and the last MLP hidden layer, for the most separated 6 classes (out of 20) from 20NG dataset. The neuron projection colors show the neurons power to discriminate each class.

The open problem, which we will try to solve in this project, is how to interpret the ANN decision in the case when data are the paragraph vectors with meaningless features. Can it be done through visual exploration of NNs used for their generation?

Datasets We will perform analysis on the SOTA datasets presented in Table 2. The main properties of each baseline dataset are: the number of samples M, dimensionality N, and the number of classes K. Here, we consider datasets with (1) a huge number of samples and relatively low dimensionality, (2) a smaller number of samples but larger number of features, (3) highly imbalanced data (RCV-Reuters), and (4) skewed data (SmallNORB). Furthermore, we will also visualize a novel dataset provided by the Warsaw Air Force Institute of Technology, which contains pictures of the fragments of aircraft fuselages with various stages of damage (rust). We will uilize *pic2vec* [53] methodology and also, we will visualize *MLP* layers obtained during training on this dataset.

Table 2: The list of baseline datasets.

Dataset	N	M	K	Short description
MNIST	784	70 000	10	Well balanced set of grayscale images of handwritten digits.
F-MNIST	784	70 000	10	More difficult MNIST version. Instead of handwritten digits it consists of apparel images.
SmallNORB	2048	48 600	5	It contains stereo image pairs of 50 uniform-colored toys under 18 azimuths, 9 elevations, and 6 lighting conditions.
RCV-Reuters	30	804 409	8	Corpus of press articles preprocessed to 30D by PCA.
YAHOO	100	1.4 million	10	Questions and answers from YAHOO. The answers service preprocessed with FastText [48].
Aviation	-	Over 0.5 million	2	It contains the fragments of aircraft fuselage images of size 480x640 pixels (rusty or clean). Dataset provided by the Warsaw Air Force Institute of Technology.

3 Related work

Machine learning technology have been developing rapidly in recent years, with possibilities growing in tandem with a greater availability of data and advancements in computing capability and storage solutions. In fact, if you look behind the scenes, you can spot many examples of machine learning technology already in practice in all kinds of industries—ranging from consumer goods and social media to financial services and manufacturing. Because of that - experts have developed many strategies to improve ANNs. During training, a common approach is comparing model accuracy on a validation set with accuracy on a training set. This helps diagnosing overfitting (low validation accuracy when compared to training accuracy) and underfitting (low accuracy in both cases). The high dimensionality of data makes them very hard to interpret, which cause lack of progress in terms of model adjustment and improvement.

Visual analytics and information visualization systems have been developed to inspect Artificial Neural Networks from the inside, since visual feedback is highly valuable. For instance, Zeiler and Fergus [59] show how insight gained from visualizing ANNs has enabled them to outperform the SOTA (at the time) on an major image classification benchmark. Their work reconstructed an input image given a particular output channel of a convolutional layer. Reconstruction from activations is also investigated by Mahendran and Vedaldi [33]. We would like to create an approach that is complementary to these visualizations, but for unstructured data representations.

Dimensionality reduction Overall, dimensionality reduction methods can be seen on a spectrum, from conceptually and computationally the simplest to the most complex and challenging. At one end of the spectrum we have linear methods such as PCA, LDS, MDS [8, 43]. PCA was invented in 1900's by Karl Pearson as an analogue of the principal axis theorem in mechanics and apart from its intrinsic usefulness is interesting because it serves as a starting point for many modern algorithms (kernel PCA [52], probabilistic PCA [56], and oriented PCA [6]). On the other end of the spectrum, there are non-linear methods, such as manifold learning methods (Laplacian eigenmaps, Isomap, UMAP), LLE, t-SNE and spectral clustering [5, 26, 32, 34, 40, 49]. Dimensionality reduction has been previously applied to ANN visualization, due to its scalability in number of dimensions and observations. For instance, Erhan et al. [14] use projections of learning trajectories to study the effects of unsupervised pre-training. Each point in such a trajectory corresponds to the concatenation of output layer activations for a whole dataset at a given training stage. In recent years many novel algorithms were presented to the public due to increasing amounts of data and advances in hardware to store and query such datasets [2, 24, 30, 45].

In the end of 2018 a manifold learning technique for dimension reduction UMAP [34] was developed. It provides a theoretical framework based in Riemmanian geometry and algebraic topology. It was build upon mathematical foundations related to the work of Belkin and Niyogi on the Laplacian Eigenmaps [5]. UMAP address the issue of uniform data distributions on manifolds. As stated in [34] method is capable of generating competitive visualizations with t-SNE [32] and preservers more of the global structure with superior run time performance, which allow it to scale to significantly larger datasets. In 2019 due to data embedding became even more pressing issue in ML many new dimensionality reduction algorithms were created. First one is TriMap [2] algorithm, which utilize triplet constraints that preservers the global accuracy of the data better than the other commonly use methods, such as t-SNE and UMAP. To quantify the global accuracy, the Authors introduced a score which roughly reflects the relative placement of the clusters rather than the individual points (it measures closeness of a given embedding in respect to the PCA embedding, which is considered to be optimal by means of preserving the data variance). Results presented by the authors showed, that TriMap in fact is better at uncovering the global structure of data, while t-SNE could provide additional insight about the local neighborhood of individual points. Next algorithm developed in 2019 is Anchor t-SNE (AtSNE) [15]. It provides an efficient GPU-based solution for large-scale and high-dimensional data, which is based on SNE concept. The main mechanism which guide the embedding process is generating so called anchor points from the original data and regard them as the skeleton of the layout. In this way, Authors preserve the global structure information. To optimize positions of the anchor points and ordinary data points in the layout a hierarchical optimization approach was employed (presented in details in [15]). Following year 2019, the Capacity Preserving Mapping (CPM) [57] technique was presented in the middle of

it. It presents a rigorous mathematical treatment to the crowding issue, which is common while high-dimensional data are being projected down to low dimensions (for visualization purposes). To do that, Authors propose a way to adjust the capacity of the high dimensional body before the dimension reduction, which results in better preservation of geometrical structure of the dataset and does not presume the existence of clusters. In this work, Authors propose: 1) a way to define a new dimension-aware distance to treat the crowding issue and 2) a method to compute the multiscale intrinsic dimensionality of a dataset that is necessary for the definition of this new distance. In January 2020 the ShapeVis [24] method was presented to the public. It relies on finding a subset of points called landmarks along the data manifold to construct a weighted witness-graph over it. This graph captures the structural characteristics of the point cloud, and its weights are determined using a Finite Markov Chain. Authors further compress this graph by applying induced maps from standard community detection algorithms. Using techniques borrowed from the manifold learning, authors prune and reinstate edges in the induced graph based on their modularity to summarize the shape of data.

Of course, the methods presented above do not exhaust the list of all created algorithms during recent years. It shows the direction in which the field of dimensionality reduction is heading and how critical it has become to have a scalable and robust systems for analyzing big data.

4 Related projects

Creation of this project is motivated by the results obtained during execution of the OPUS research project "Interactive Visual Text Analytics (IVTA): Development of novel, user-driven text mining and visualization methods for large text corpora exploration" (NCN 2013/09/B/ST6/01549, 2013-2016) in which Professor dr Witold Dzwinel was the Principal Investigator. The main scientific hypothesis of that project was that visualization and interactive exploration by an expert is an efficient way for incorporating his prior knowledge and data exploration experience in development of models for text large text corpora. The main project goal was to provide a novel framework – Interactive Visual Text Analytics (IVTA) – which integrates new data visualization methods and novel methodologies for incorporating both prior "expert knowledge" and posterior knowledge acquired during the process of interactive data exploration. The outcome of the IVTA project are 29 publications (including 8 JCR papers, and 5 papers presented on international conferences of CORE A and A* rank) and 2 completed PhD theses.

The project that we propose here will use both the software tools and the expertise acquired during execution of the IVTA project. However, the current project is independent and does not represent a direct extension of the previous one. The problems and tasks formulated in this proposal regarding visualization of high-dimensional data and visualizing the hidden activity of Artificial Neural Network are novel, and fall in the mainstream of current scientific challenges in modern machine learning.

5 Implementation

Research team. The project will be carried out by the research team from the Department of Computer Science, AGH University of Science and Technology (Faculty of Computer Science, Electronics and Telecommunications), which consists of 2 faculty researchers: one full professor (Witold Dzwinel[2], PhD) and a PhD student (Bartosz Minch, MSc). The expertise of the team is:

- exceptional in: scientific visualization, modeling and simulation, high performance computing, parallel computing, GPU computing,
- very strong in: machine learning and pattern recognition, including DNNs and ensemble data models,

[2] https://www.researchgate.net/profile/Witold_Dzwinel

- strong in: implementing libraries, efficient algorithms, web services, designing system architectures (OOP),
- good in: GPU/CUDA programming, physics,

Table 3: Bibliometric data of the research team. Data reported from Scopus and Google Scholar (in brackets).

Affiliation	Team Members	H-index	#Citations
AGH Univeristy of Science and Technology	Witold Dzwinel[2]	19(25)	1022(1902)
AGH Univeristy of Science and Technology	Bartosz Minch (PhD Student)	1(1)	2(2)

There is also possibility to use a full expertise of complex systems team, which is led by Professor dr Witold Dzwinel[2]. Such a cooperation with experts in the field of machine learning and complex systems will ensure highly rigorous level of the research, and access to cutting edge algorithms and concepts.

Resources to be committed. The project requires access to computational power and other computational facilities (such as software and support). In this context the key assets of the applicant is the availability of high performance computing resources in the Academic Computer Centre CYFRONET AGH [3], and in particular an access to PROMETHEUS, the fastest supercomputer in Poland. PROMETHEUS provides, among other, more than a hundred GPU units, including fast Tesla V100 cards. Other important assets are computational and visualization facilities in the AGH Department of Computer Science [4]. The applicant has an experience in embedding large datasets and training neural networks on the PROMETHEUS supercomputer, and believes that its computational resources are more than sufficient for the requirements of this project. In addition to the mentioned hardware resources, the researchers has experience in common data embedding/machine learning frameworks. Furthermore, it develops specialized software for visualization of very large high-dimensional datasets. In summary, the author of this proposal, have all resources required for successful realization of the project of this type and size.

6 Expected project results

We will strive to publish our results mainly in top Machine Learning and Artificial Intelligence conferences (CORE A*), such as: ICML, NeurIPS (formerly NIPS), ACL, AAAI, IJCAI, KDD. However, knowing the selectivity of these conferences we will also consider lower rank conferences such as: ECML, EMNLP, ESANN, ICNN, IWANN, ANNIE. We also plan to submit papers to ICR journals such as: Journal of Machine Learning Research, Machine Learning, IEEE Transactions on Neural Networks, Neural Networks, Journal of Visualization, Neurocomputing and others from ML and AI domains. In addition, we expect to further produce original pieces of software, which will be publicly available in the form of open source repository. Particularly, in the scope of this project we plan to develop original software for visual analysis that would be useful for very large datasets.

[2] https://www.researchgate.net/profile/Witold_Dzwinel

[3] http://www.cyfronet.krakow.pl/13243,artykul,computers.html

[4] https://www.informatyka.agh.edu.pl/en/research-development/laboratories/

References

[1] Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). 1–6 (2017)

[2] Amid, E., Warmuth, M.K.: TriMap: Large-scale Dimensionality Reduction Using Triplets. arXiv preprint arXiv:1910.00204 (2019)

[3] Bae, S.H., Qiu, J., Fox, G.C.: High performance multidimensional scaling for large high-dimensional data visualization. In: IEEE Transaction of Parallel and Distributed System (2012)

[4] Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15**(6), 1373–1396 (2003)

[5] Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems 14, 585–591. MIT Press (2002)

[6] Bermejo, S., Cabestany, J.: Oriented principal component analysis for large margin classifiers. Neural Networks **14**(10), 1447 – 1461 (2001)

[7] Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American Statistical Association **112**(518), 859–877 (2017)

[8] Buja, A., Swayne, D., Littman, M., Dean, N., Heike, H., Chen, L.: Data visualization with multidimensional scaling. Journal of Computational and Graphical Statistics **17**, 444–472 (2008)

[9] Dogo, E.M., Afolabi, O.J., Nwulu, N.I., Twala, B., Aigbavboa, C.O.: A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In: 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS). 92–99 (2018)

[10] Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**, 2121–2159 (2011)

[11] Dzwinel, W., Blasiak, J.: Method of particles in visual clustering of multi-dimensional and large data sets. Future Generation Comp. Syst. **15**, 365–379 (1999)

[12] Dzwinel, W., Wcislo, R., Matwin, S.: 2-d embedding of large and high-dimensional data with minimal memory and computational time requirements. arXiv preprint arXiv:1902.01108 (2019)

[13] Dzwinel, W., Wcislo, R., Strzoda, M.: ivga: visualization of the network of historical events. In: Proceedings of the 1st International Conference on Internet of Things and Machine Learning. 1–7 (2017)

[14] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. **11**, 625–660 (2010)

[15] Fu, C., Zhang, Y., Cai, D., Ren, X.: Atsne: Efficient and robust visualization on gpu through hierarchical optimization. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 176–186 (2019)

[16] Goldberg, Y., Zakai, A., Kushnir, D., Ritov, Y.: Manifold learning: The price of normalization. arXiv preprint arXiv:0806.2646 (2008)

[17] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. arXiv preprint arXiv:1607.00653 (2016)

[18] Hinton, G.E., Roweis, S.T.: Stochastic neighbor embedding. In: Advances in neural information processing systems. 857–864 (2003)

[19] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**, 1735–80 (1997)

[20] Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. arXiv preprint arXiv:1608.06993 (2016)

[21] Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 (2017)

[22] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[23] Kłusek, A., Dzwinel, W.: Multi-gpu k-nearest neighbor search in the context of data embedding. Advances in Parallel Computing (2017)

[24] Kumari, N., R., S., Rupela, A., Gupta, P., Krishnamurthy, B.: Shapevis: High-dimensional data visualization at scale. arXiv preprint arXiv:2001.05166 (2020)

[25] Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. arXiv preprint arXiv:1405.4053 (2014)

[26] Lee, J.A., Verleysen, M.: Nonlinear dimensionality reduction of data manifolds with essential loops. Neurocomput. **67**, 29–53 (2005)

[27] Li, Z., Shi, W., Shi, X., Zhong, Z.: A supervised manifold learning method. Comput. Sci. Inf. Syst. **6**, 205–215 (12 2009)

[28] Linderman, G., Rachh, M., Hoskins, J.: Efficient algorithms for t-distributed stochastic neighborhood embedding. arXiv preprint arXiv:1712.09005 (2017)

[29] Liu, R., Wang, H., Yu, X.: Shared-nearest-neighbor-based clustering by fast search and find of density peaks. Information Sciences **450**, 200 – 226 (2018)

[30] Lu, Y., Corander, J., Yang, Z.: Doubly stochastic neighbor embedding on spheres. arXiv preprint arXiv:1609.01977 (2016)

[31] van der Maaten, L.: Accelerating t-sne using tree-based algorithms. Journal of Machine Learning Research **15**, 3221–3245 (2014)

[32] van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)

[33] Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. arXiv preprint arXiv:1412.0035 (2014)

[34] McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)

[35] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

[36] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26, 3111–3119. Curran Associates, Inc. (2013)

[37] Minch, B., Nowak, M., Wcislo, R., Dzwinel, W.: Gpu-embedding of knn-graph representing large and high-dimensional data. ICCS 2020 preprint (2020)

[38] Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: Learning distributed representations of graphs. arXiv preprint arXiv:1707.05005 (2017)

[39] Nesterov, Y.: A method for unconstrained convex minimization problem with the rate of convergence $o((1/k)^2)$. Doklady AN USSR **269**, 543–547 (1983)

[40] Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. 849–856. MIT Press (2001)

[41] Nowak, M.: Ivhd-cuda (2019), [Online; accessed 15-May-2020]

[42] Pawliczek, P., Dzwinel, W.: Interactive data mining by using multidimensional scaling. Procedia Computer Science **18**, 40 – 49 (2013)

[43] Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine **2**, 559–572 (1901)

[44] Pezzotti, N., Höllt, T., Lelieveldt, B., Eisemann, E., Vilanova, A.: Hierarchical stochastic neighbor embedding. Computer Graphics Forum **35**, 21–30 (2016)

[45] Pezzotti, N., Höllt, T., Lelieveldt, B., Eisemann, E., Vilanova, A.: Hierarchical stochastic neighbor embedding. Computer Graphics Forum **35**, 21–30 (2016)

[46] Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., Carin, L.: Variational autoencoder for deep learning of images, labels and captions. arXiv preprint arXiv:1609.08976 (2016)

[47] Rauber, P.E., Fadel, S.G., Falcão, A.X., Telea, A.C.: Visualizing the hidden activity of artificial neural networks. IEEE Transactions on Visualization and Computer Graphics **23**(1), 101–110 (2017)

[48] Research, F.A.: Library for fast text representation and classification. (2018), [Online; accessed 20-November-2019]

[49] Ribeiro, B., Vieira, A., Carvalho das Neves, J.: Supervised isomap with dissimilarity measures in embedding learning. In: Progress in Pattern Recognition, Image Analysis and Applications. 389–396. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)

[50] Rocca, J.: Understanding variational autoencoders (vaes) (2019), [Online; accessed 21-May-2020]

[51] Scholz, M., Fraunholz, M., Selbig, J.: Nonlinear principal component analysis: Neural network models and applications. In: Principal Manifolds for Data Visualization and Dimension Reduction. 44–67. Springer Berlin Heidelberg (2008)

[52] Schölkopf, B., Smola, A., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10**(5), 1299–1319 (1998)

[53] Singla, K., Mukherjee, N., Koduvely, H.M., Bose, J.: Evaluating usage of images for app classification. arXiv preprint arXiv:1912.12144 (2019)

[54] Tang, J., Liu, J., Zhang, M., Mei, Q.: Visualizing large-scale and high-dimensional data. In: Proceedings of the 25th International Conference on World Wide Web. 287–297 (2016)

[55] Tao, Y., Papadias, D., Lian, X.: Reverse knn search in arbitrary dimensionality. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30. 744–755 (2004)

[56] Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B **61**(3), 611–622 (1999)

[57] Wang, R., Zhang, X.: Capacity preserving mapping for high-dimensional data visualization. arXiv preprint arXiv:1909.13322 (2019)

[58] Zeiler, M.D.: Adadelta: An adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)

[59] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. arXiv preprint arXiv:1311.2901 (2013)