JMLR: Workshop and Conference Proceedings 39:221–233, 2014 ACML 2014

Nonlinear Dimensionality Reduction of Data by Deep Distributed Random Samplings

Xiao-Lei Zhang HUOSHAN6@126.COM
Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China, 100084.

Editor: Dinh Phung and Hang Li

Abstract

Dimensionality reduction is a fundamental problem of machine learning, and has been intensively studied, where classification and clustering are two special cases of dimensionality reduction that reduce high-dimensional data to discrete points. Here we describe a simple multilayer network for dimensionality reduction that each layer of the network is a group of mutually independent k-centers clusterings. We find that the network can be trained successfully layer-by-layer by simply assigning the centers of each clustering by randomly sampled data points from the input. Our results show that the described simple method outperformed 7 well-known dimensionality reduction methods on both very small-scale biomedical data and large-scale image and document data, with less training time than multilayer neural networks on large-scale data.

Keywords: Bootstrap, deep learning, dimensionality reduction, ensemble methods, evolutionary computing, kernel methods, sparse coding.

1. Introduction

An excellent intelligent machine can at least learn the basic semantics of an objective, such as the sources of a speech or image separation problem, the disease types of a patient's DNA sequence, the words of a handwritten or spoken sentence, the topics or sentiment of a story, without interfered by small variations of the objective. A common learning method is dimensionality reduction. The vitality of a dimensionality reduction method depends not only on its performance but also on how easily people in different areas can understand it, implement it, and use it. A simple and widely used method is principle component analysis (PCA), which aims to find a coordinate system that the linearly uncorrelated coordinate variables (called principle components) describe the most variances of data. Because PCA is insufficient to capture highly-nonlinear data distributions, many nonlinear dimensionality reduction methods have been proposed.

Among the nonlinear methods, one prevalent class of nonlinear dimensionality reduction methods are the nonparametric graph models (Schölkopf et al. (1997); Shi and Malik (2000); Tenenbaum et al. (2000); Roweis and Saul (2000); Ng et al. (2002); Belkin and Niyogi (2003); He and Niyogi (2004); Yan et al. (2007); Van der Maaten and Hinton (2008)), which mainly try to keep the pairwise similarities of data points in the low-dimensional space as similar as possible as those in the original high-dimensional space. Because they need to calculate each pairwise similarity, their time and storage complexities scale squarely with the number

of data points, limiting them to small-scale problems. Another typical nonlinear method is multilayer neural network (Rumelhart et al. (1986); Hinton and Salakhutdinov (2006)), which gradually reduces the dimensionality of data (i.e., learns more and more abstract features) through multiple layers of nonlinear transforms. It can handle large-scale problems well, but it is limited to large-scale problems and is difficult to be trained successfully with too many layers. Moreover, its structure and its training method (Rumelhart et al. (1986); Hinton and Salakhutdinov (2006)) are complicated and need careful manual-engineering, making it restricted to the experts of artificial intelligence.

Besides, many machine learning techniques are doing dimensionality reduction (or approximation of data distribution) by either reducing the dimensionality of data explicitly or generating sparse high-dimensional features of data implicitly, such as (hierarchical) probabilistic models (Hofmann (1999); Blei et al. (2003); Hinton et al. (2006)) and sparse coding (Olshausen and Field (1996)). See (Van der Maaten et al. (2009); Sorzano et al. (2014)) for excellent reviews of dimensionality reduction.

Here, we describe a very simple and robust multilayer network, named deep distributed random samplings (DDRS), that can reduce the dimensionality effectively and efficiently on both large-scale and small-scale problems. Methods, Extended Data figures, Supplementary Discussion and source code are available at http://sites.google.com/site/zhangxiaolei321/.

2. Algorithm

DDRS contains multiple hidden layers and an output layer (Fig. 1a). Each hidden layer is a group of mutually independent k-centers clusterings; each k-centers clustering has k output units, each of which indicates one cluster; the output units of all k-centers clusterings are concatenated as the input of their upper layer. The output layer is PCA. Parameter k should be as large as possible at the bottom layer and be smaller and smaller along with the increase of the number of layers.

DDRS is trained simply layer-by-layer. For training each layer given a d-dimensional input data set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ either from the lower layer or from the original data space, we simply need to focus on training each k-centers clustering, which consists of four steps with the third step for small-scale problems only.

- **Random feature selection.** The first step randomly selects $\hat{d}$-dimensional features of $\mathcal{X}$ ($\hat{d} \leq d$) to form a subset of $\mathcal{X}$, denoted as $\hat{\mathcal{X}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n\}$.
- **Random sampling.** The second step randomly selects k data points from $\hat{\mathcal{X}}$ as the k centers of the clustering, denoted as $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$.
- **Random reconstruction.** When k approximates to n (i.e., the problem is small-scale), the third step randomly selects d' dimensions of the k centers ($d' \leq \hat{d}/2$) and does one-step cyclic-shift as shown in Fig. 1b.
- **Sparse representation learning.** The fourth step assigns each input data point $\hat{\mathbf{x}}$ to one of the k clusters and outputs a k-dimensional indicator vector $\mathbf{h} = [h_1, \ldots, h_k]^T$ which will be part of the input feature of $\mathbf{x}$ to the upper layer as shown in Fig. 1a, where operator ^{T} denotes the transpose of vector. For example, if $\hat{\mathbf{x}}$ is assigned to the second cluster, then $\mathbf{h} = [0, 1, 0, \ldots, 0]^T$. The assignment is calculated according to

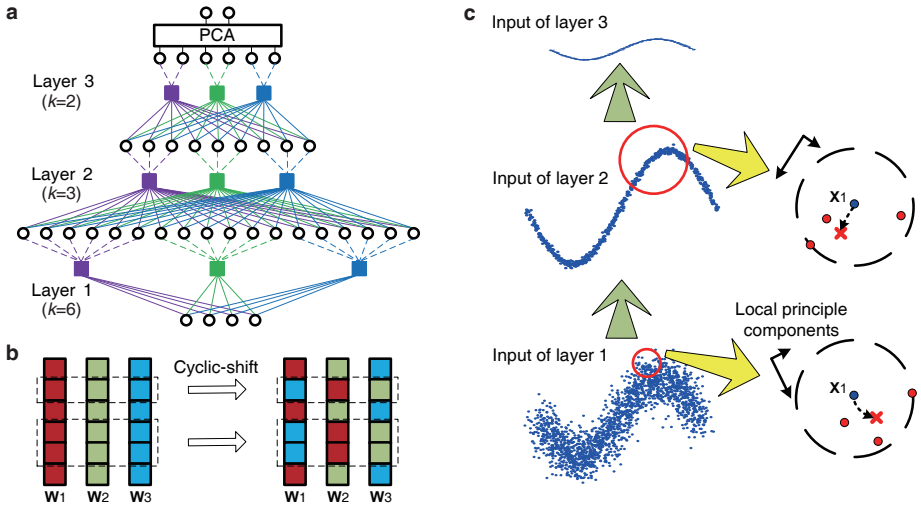

Figure 1: **Description of DDRS. a,** The DDRS network. Each square represents a k-centers clustering. **b,** Random reconstruction. The columns represent the centers of a 3-centers clustering. Each square represents an entry of a center. **c,** Principle of DDRS. The regions in the red circles represent the local areas of the data point $\mathbf{x}_1$, which are further amplified in the dashed circles. Each red point in the dashed circles is the closest center of a k-centers clustering to $\mathbf{x}_1$. The new representation of $\mathbf{x}_1$ in each layer is marked as a red cross in the dashed circle. The local principle components are shown in the upper and left corners of the dashed circles.

the similarities between $\hat{\mathbf{x}}$ and the k centers, in terms of some predefined similarity measurement at the bottom layer, such as the Euclidean distance $\arg\min_{i=1}^{k} \|\mathbf{w}_i - \hat{\mathbf{x}}\|^2$, or in terms of $\arg\max_{i=1}^{k} \mathbf{w}_i^T \hat{\mathbf{x}}$ at all other hidden layers.

DDRS handles large-scale problems well. For training each layer, the time complexity is $O\left(nsk^2V^2\right)$, and the storage complexity is $O(2nskV)$, where V is the number of clusterings and s the sparsity of the input data (i.e., the ratio of the non-zero elements over all elements); particularly, $s = 1/k$ (see Supplementary Discussion).

3. Theoretical justification

DDRS has a simple geometric explanation. It first conducts the piecewise-linear dimensionality reduction—a local PCA that gradually enlarges the local region (Fig. 1c)—implicitly in the hidden layers, and then gets low-dimensional features explicitly by PCA. Specifically, each data point (e.g., $\mathbf{x}_1$ in Fig. 1c) owns a local region supported by the centers of all clusterings that are closest to the data point. The centers define the local coordinate system. The new representation of the data point is the coordinates of the data point in the local coordinate system. If some other data points share the same local region, they will also be projected to the same coordinates, which means the small variances (i.e., small principle components) of this local region that are not covered by the local coordinate system will be

discarded. It is easy to image that when k is smaller and smaller, the local region is gradually enlarged, making larger and larger relatively-unimportant local variances discarded. However, when k approximates to n, the areas of most local regions are zero, resulting in no approximations (i.e., dimensionality reductions) in these regions. To prevent this unwanted situation, we borrowed the reconstruction step (a.k.a., crossover) of the genetic algorithm (Holland (1975)). After random reconstruction, the centers not only will not appear in the input data but also can still define the coordinate systems of the local data distributions.

Besides the geometric explanation, DDRS is also rooted in the bootstrap theory and regularization theory from the statistics and machine learning perspectives. According to the theories of bootstrap resampling (Efron (1979); Efron and Tibshirani (1993); Breiman (1996)), weak learnability (Schapire (1990)), and ensemble methods (Dietterich and Bakiri (1995); Dietterich (2000); Breiman (2001); Zhou et al. (2002); Strehl and Ghosh (2003); Fred and Jain (2005); Zhou (2012)), DDRS is a stack of bootstraps or clustering ensembles: each k-centers clustering is a bootstrap sample or a weak learner that is slightly better than random guessing; multiple clusterings group to a strong learner that reduces the variances of local regions effectively. According to the theory of regularization (Tikhonov (1963); Poggio and Girosi (1990a,b)), DDRS is a stack of regularization networks: each k-centers clustering is a ℓ_∞-norm-regularized two-layer network; motivated by the relationship between adaptive boosting and support vector machine on Vapnik-Chervonenkis dimension (Schapire et al. (1998); Freund and Schapire (1995); Cortes and Vapnik (1995); Vapnik (1998)), we proved that learning a group of k-centers clusterings is a sparse coding that is lower bounded by the ℓ_1-norm-regularized sparse coding (Olshausen and Field (1996)). See Supplementary Discussion for the detailed explanation.

DDRS was motivated from the weakness of DBN (Hinton et al. (2006); Hinton and Salakhutdinov (2006)) and was developed step-by-step as follows. (i) It is known that DBN lacks the *explaining away* property (Bengio et al. (2013); Bengio (2009)). In order to own the explaining away property, we generalized the most compact *product of experts* (PoE) (Hinton (1999, 2002)) that each expert owns only one hidden unit to a PoE that each expert owns multiple hidden units. (ii) Motivated by the factorization and fast training method of restricted Boltzmann machine (Hinton et al. (2006); Hinton and Salakhutdinov (2006)) (the building block of DBN), we proposed to train each expert independently via the expectation-maximization optimization (Bishop et al. (2006)), which is formulated as learning a group of k-means clusterings. (iii) Motivated by the contrastive divergence (CD) learning (Hinton (2002)), which is a t-step ($t \geq 1$) approximation of maximum likelihood learning, we proposed to discard the expectation-maximization optimization but only preserve the default initialization method—random sampling for modeling the k-centers. (iv) Because random sampling on small-scale data sets will cause overfitting, we borrowed the reconstruction step of the genetic algorithm to further process the selected data points. (v) DDRS is also related to convolution neural networks, sparse coding, ensemble learning, and manifold learning. See Supplementary Discussion for the detailed explanation.

4. Empirical study

To demonstrate the effectiveness of DDRS on small-scale data sets, we compared DDRS with PCA and 4 well-known nonlinear dimensionality reduction methods, including isomet-

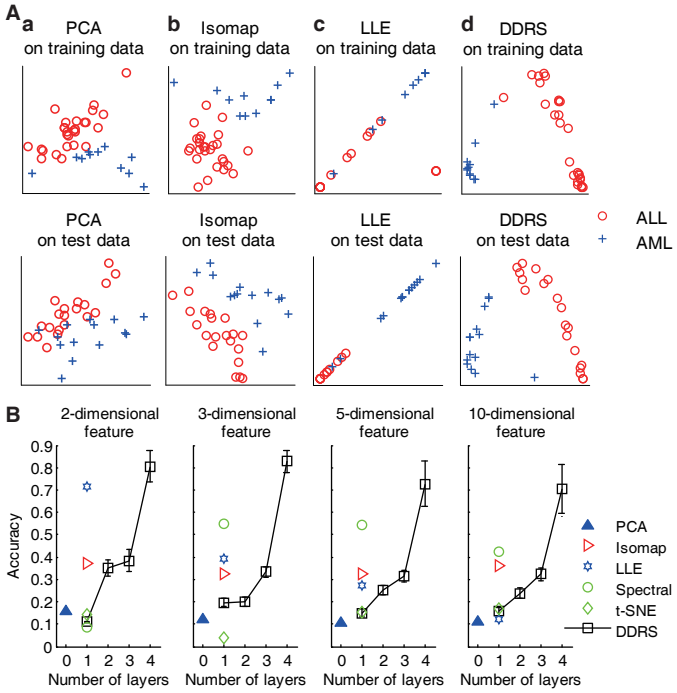

Figure 2: **Reducing the dimensionality of the 7,129-dimensional AML-ALL biomedical data which consist of 72 examples. A, a-d,** Visualizations produced by PCA, Isomap, LLE, and DDRS at layer 4 respectively. The visualizations produced by other layers of DDRS are shown in Extended Data Fig. 1. **B,** Accuracy comparison of the k-means clusterings using the low-dimensional features produced by DDRS and 5 competitive methods respectively.

ric feature mapping (Isomap) (Tenenbaum et al. (2000)), locally linear embedding (LLE) (Roweis and Saul (2000)), spectral clustering (Spectral) (Ng et al. (2002)), and t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten and Hinton (2008)), on the acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) biomedical data which consist of only 38 training examples and 34 test examples (Golub et al. (1999)) (see Methods). Because multilayer neural network cannot handle such a small-scale data set, we will not compare with it. We produced low-dimensional features of the data set for visualization (Fig. 2A and Extended Data Fig. 1) and clustering (Fig. 2B). The clustering accuracy was measured by the normalized mutual information (Strehl and Ghosh (2003)). Experimental results show that DDRS outperformed the competitive methods on both visualization and clustering accuracy.

We also compared DDRS with the aforementioned 5 competitive methods and deep belief network (DBN) (Hinton and Salakhutdinov (2006)), a multilayer neural network with a special stochastic initialization method, on small subsets of the MNIST handwritten digits (Lecun et al. (2004)), each of which consists of 5,000 images (see Methods). Experimental results show that (i) DDRS achieved an ideal visualization of the 10 digits and outperformed

other competitive methods; (ii) although t-SNE also achieved a clear visualization, the visualization produced by DDRS had larger between-class distances and smaller within-class variations (i.e., clearer visualization when we do not draw colors on the digits) than that produced by t-SNE (Fig. 3 and Extended Data Fig. 2). When the low-dimensional features were applied for clustering, DDRS outperformed the competitive methods, and the highest accuracy of DDRS was over 80% (Fig. 4).

To investigate the scalability and effectiveness of DDRS on large-scale problems, we compared DDRS with PCA and DBN on the MNIST handwritten digits (Lecun et al. (2004)) which consist of 60,000 training images and 10,000 test images (see Methods). Because Isomap, LLE, Spectral, and t-SNE cannot handle such a large-scale problem, we did not compare with them. Experimental results show that DDRS achieved a better visualization than DBN (figure 3B in ref. (Hinton and Salakhutdinov (2006))) on the 10 digits, such as the digital pair "3, 5, and 8" (Fig. 5a and Extended Data Fig. 3). When the low-dimensional features were applied for clustering, DDRS was as good as DBN when they had the same number of layers, and outperformed DBN when more layers were easily stacked (Fig. 5b). When the experiment was run on a one-core computer, DDRS consumed one-order less training time than DBN (Fig. 5c).

We also compared DDRS with DBN and latent semantic analysis (LSA) (Deerwester et al. (1990)), a well-known document retrieval method based on PCA, on a larger data set—Reuters newswire stories (Lewis (2004)) which consist of 804,414 documents with half of the documents used for training and the other half for test (see Methods). Experimental results show that DDRS achieved a better visualization than DBN (figure 4C in ref. (Hinton and Salakhutdinov (2006))) on the 9 demo topics, such as the topics "European Community" and "moneytary/economic" (Fig. 6a and Extended Data Fig. 4). When the documents were reduced to five-dimensional features for document retrieval, DDRS reached an accuracy of about 10% higher than LSA when only a handful of documents were retrieved, and this superiority was enlarged slightly when more documents were retrieved (Fig. 6b). When the experiment was run on a one-core personal computer, DDRS was faster than DBN on the training time (Fig. 6c).

Besides the above empirical comparisons, we further analyzed the impacts of different parameter settings of DDRS on the performance (see Methods). Experimental results show that (i) fortunately, the time complexity of DDRS scaled linearly but not squarely with V, which can only be explained as that the input features of each layer were sparse (Extended Data Fig. 5); (ii) the performance of DDRS was robust to the parameter selection (Extended Data Figs. 6 to 9).

5. Concluding remarks

In this paper, we have proposed a new deep (i.e., multilayer) network for nonlinear dimensionality reduction—DDRS. DDRS has a novel network structure that each expert in a layer is a k-centers clustering. The k centers of each expert is only k randomly sampled data points from the training data. For small-scale problems, the k centers are reconstructed by a simple cyclic-shift operation. The time and storage complexities of DDRS scale linearly (but not quadratically) with the size of the training data. Moreover, it is quite easily understood, implemented, and used, even without the knowledge of machine learning. It

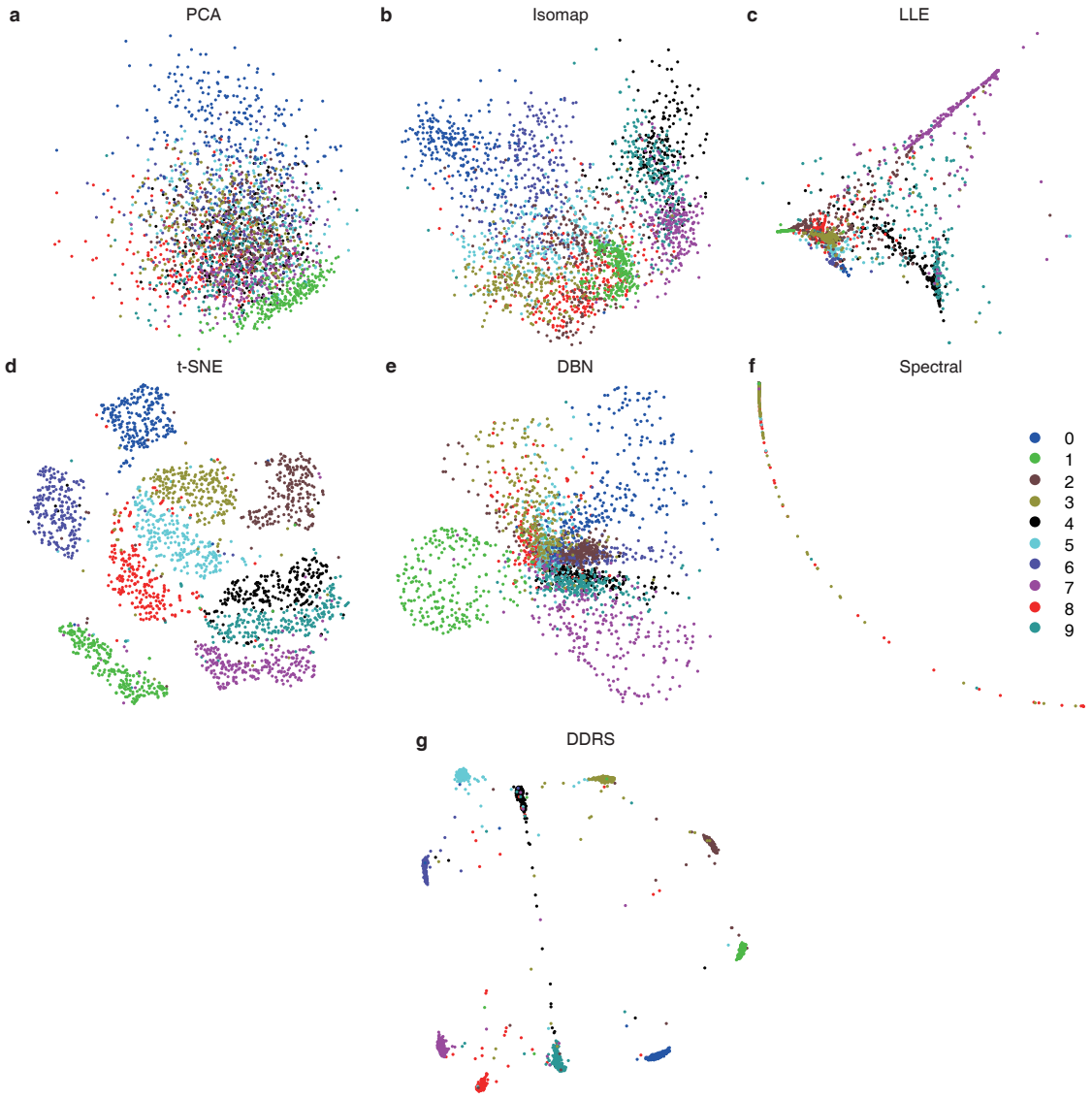

Figure 3: **Reducing the dimensionality of subsets of the 784-dimensional MNIST handwritten digits, each of which consists of 5,000 images for visualization. a-g,** Visualizations produced by 6 competitive methods and DDRS at layer 7. The visualizations produced by other layers of DDRS are shown in Extended Data Fig. 2.

performs robustly with different parameter settings. It can support distributed computing naturally. Empirical results have shown that DDRS can learn more and more abstract representations successfully on both large-scale and small-scale problems with less training time than DBN on large-scale problems.

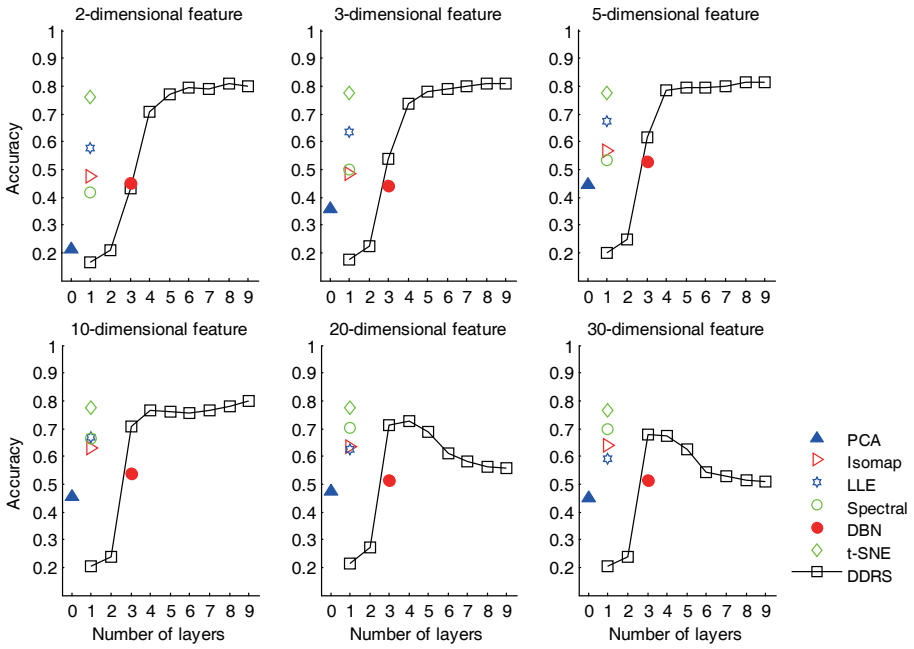

Figure 4: **Reducing the dimensionality of subsets of the 784-dimensional MNIST handwritten digits, each of which consists of 5,000 images, for clustering.** Accuracy comparison of the k-means clusterings using the low-dimensional features produced by DDRS and 6 competitive methods respectively.

Theoretically, DDRS extended bootstrap resampling methods (Efron (1979); Efron and Tibshirani (1993); Breiman (1996); Schapire (1990); Freund and Schapire (1995)) and kernel methods (Poggio and Girosi (1990a,b); Cortes and Vapnik (1995); Vapnik (1998)) to the unsupervised deep architecture, borrowed evolutionary computing (Holland (1975)) to prevent the overfitting problem, and made kernel methods competitive on large-scale problems. With the recent big explosion of data and fast development of computing power, it was once thought that kernel methods cannot compete with neural networks, since they have squared time and storage complexities. Our results show that kernel methods could still be highly competitive with neural networks on large-scale problems by simply stacking multiple sub-samplings of the columns of the kernel matrix without calculating the entire kernel matrix.

Compared to learning with kernels (Cortes and Vapnik (1995); Vapnik (1998); Schölkopf and Smola (2002); Schölkopf et al. (1997); Shi and Malik (2000); Ng et al. (2002)), DDRS scales linearly with the size of the dataset, which overcomes the fundamental weaknesses of learning with kernels (or graphs). Compared to traditional neural networks which minimizes the empirical risk with many bad local minima (Vapnik (1998)), DDRS minimizes the structural risk (Vapnik (1998)) under the smooth assumption that a small variation on the input data results in only a small variation on the output target (Poggio and Girosi (1990a)), such that it performs well not only on large-scale problems but also on very small-

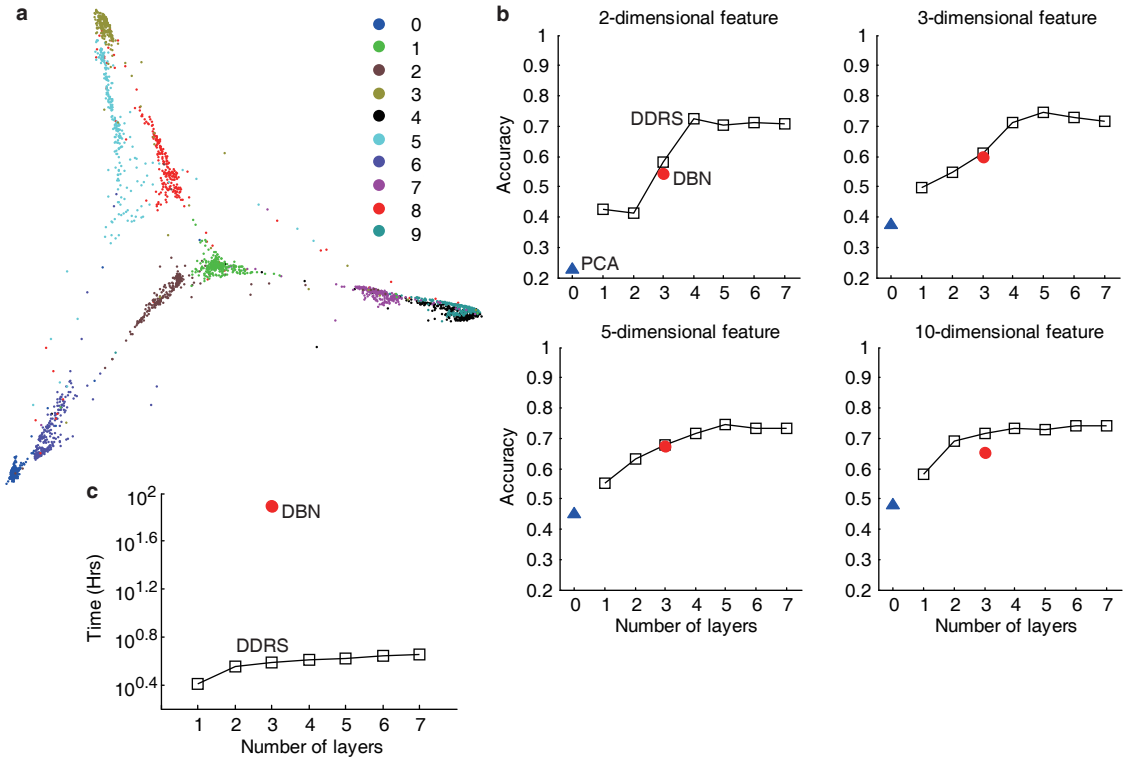

Figure 5: **Reducing the dimensionality of the 784-dimensional MNIST handwritten digits which consist of 70,000 images. a,** Visualization of MNIST produced by DDRS at layer 7. For clarity, only 500 images per digit are drawn. The visualizations produced by other layers of DDRS are shown in Extended Data Fig. 3. **b,** Accuracy comparison of the k-means clusterings using the low-dimensional features produced by DDRS, PCA and DBN respectively on the 10,000 test images. **c,** Training time (in hours) comparison between DDRS and DBN on MNIST.

scale problems. As a word, DDRS, as a stack of unsupervised bootstraps, integrates the generalization ability of learning with kernels and the scalability of neural networks.

Acknowledgments

The author thanks Prof. DeLiang Wang for providing the Ohio Supercomputing Center, Columbus, OH, USA for the empirical study.

References

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

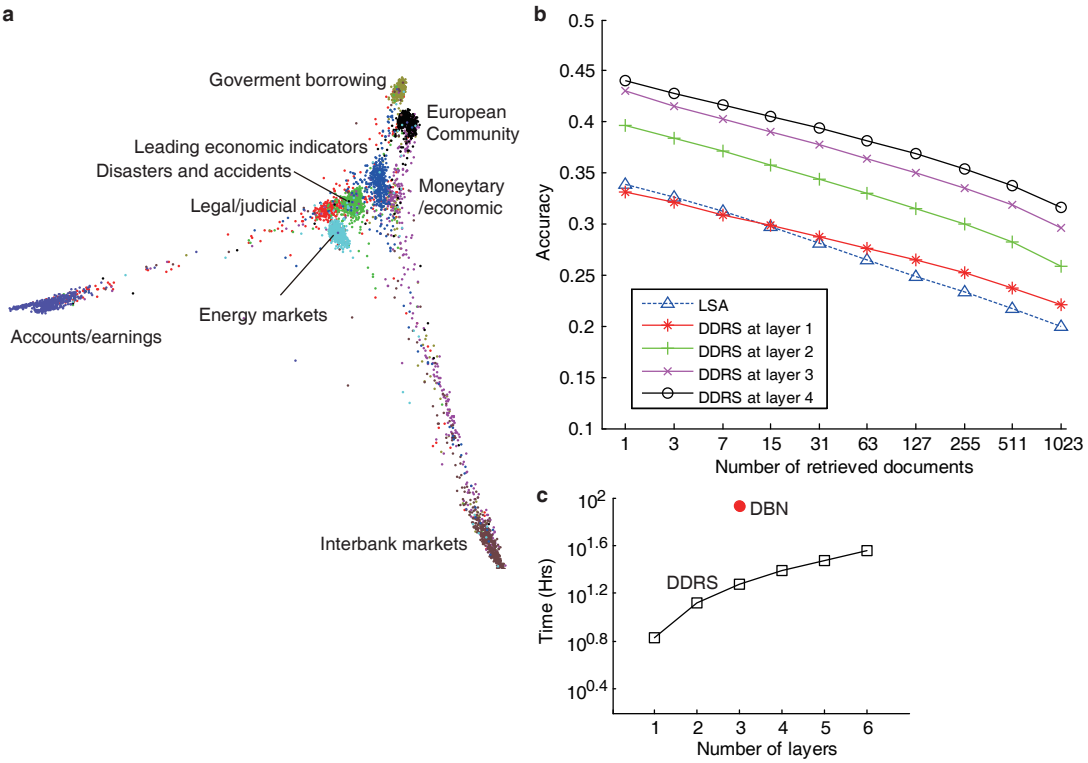

Figure 6: **Reducing the dimensionality of the 2,000-dimensional Reuters newswire stories which consist of 804,414 documents. a,** Visualization of 9 demo topics of the Reuters newswire stories produced by DDRS at layer 6. For clarity, only 500 documents per topic are drawn. The visualizations produced by other layers of DDRS are shown in Extended Data Fig. 4. **b,** Average accuracy curves of retrieved documents over 402,207 queries (including 82 topics) on the test set of the Reuters newswire stories produced by DDRS and LSA. Each query is a 5-dimensional document from the test set. The accuracy is defined as the proportion of the retrieved documents that are in the same class as the query. **c,** Training time (in hours) comparison between DDRS and DBN.

Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Maching Intell.*, 35(8):1798–1828, 2013.

Christopher M Bishop et al. *Pattern Recognition and Machine Learning*. Springer New York:, 2006.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

Leo Breiman. Bagging predictors. *Machine Learn.*, 24(2):123–140, 1996.

Leo Breiman. Random forests. *Machine Learn.*, 45(1):5–32, 2001.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learn.*, 20(3): 273–297, 1995.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.*, 41 (6):391–407, 1990.

Thomas G Dietterich. Ensemble methods in machine learning. *Multiple Classifier Systems*, pages 1–15, 2000.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *J. Artif. Intell. Res.*, 2:263–286, 1995.

Bradley Efron. Bootstrap methods: another look at the jackknife. *Ann. Stat.*, 7(1):1–26, 1979.

Bradley Efron and Robert Tibshirani. *An Introduction to the Bootstrap*. CRC press, 1993.

Ana LN Fred and Anil K Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(6):835–850, 2005.

Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the 2nd European Conference on Computational Learning Theory*, pages 23–37, Barcelona, Spain, 1995.

Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43, 1999.

Xiaofei He and X Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 17*, volume 16, pages 153–160, Vancouver, British Columbia, Canada, 2004.

Geoffrey E Hinton. Products of experts. In *Proceedings of the 9th International Conference on Artificial Neural Networks*, pages 1–6, Edinburgh, United Kingdom, 1999.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, 2002.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, 2006.

Thomas Hofmann. Probabilistic latent semantic indexing. In *Proc. 22nd Int. ACM SIGIR Conf. Res. Dev. Inform. Retrieval*, pages 50–57. ACM, 1999.

John H Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* U Michigan Press, 1975.

Yann Lecun, Corinna Cortes, and C. J. C. Burges. THE MNIST DATABASE of handwritten digits. http://yann.lecun.com/exdb/mnist/index.html, 2004.

David D. Lewis. RCV1-v2/LYRL2004: The LYRL2004 Distribution of the RCV1-v2 Text Categorization Test Collection. http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm, 2004.

Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, Vancouver, British Columbia, Canada, 2002.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proc. IEEE*, 78(9):1481–1497, 1990a.

Tomaso Poggio and Federico Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247(4945):978–982, 1990b. doi: 10.1126/science.247.4945.978.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Robert E Schapire. The strength of weak learnability. *Machine Learn.*, 5(2):197–227, 1990.

Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.*, 26(5):1651–1686, 1998.

Bernhard Schölkopf and Alexander J Smola. *Learning With Kernels.* The MIT Press, 2002.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Proceedings of the 7th International Conference on Artificial Neural Networks*, pages 583–588, Lausanne, Switzerland, 1997.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8):888–905, 2000.

COS Sorzano, J Vargas, and A Pascual Montano. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*, pages 1–35, 2014.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Andrey Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 5:1035–1038, 1963.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9(85):2579–2605, 2008.

LJP Van der Maaten, EO Postma, and HJ Van Den Herik. Dimensionality reduction: A comparative review. *J. Mach. Learn. Res.*, 10:1–41, 2009.

Vladimir N Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(1):40–51, 2007.

Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.

Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artif. Intell.*, 137(1):239–263, 2002.