Nonlinear Dimensionality Reduction using Neural Networks

Ruslan Salakhutdinov
(Joint work with Geoff Hinton)
Department of Computer Science
University of Toronto
rsalakhu,hinton@cs.toronto.edu

1 Extended Abstract

Discovering low-dimensional structure from high-dimensional observations has always been an important task in machine learning. The compact representation can be used for exploratory data analysis, preprocessing, and data visualization. There exist a variety of dimensionality reduction techniques, which can be broadly classified into: Linear methods (such as Principal Component Analysis (PCA)), Non-linear mappings (such as autoencoders), and Proximity based methods (such as Local Linear Embedding (LLE) [6]).

Most of the existing algorithms suffer from various drawbacks. If the data lie on an embedded low-dimensional nonlinear manifold, then linear methods, even though computationally efficient, cannot recover this structure. Proximity based methods are more powerful, but their computational cost scales quadratically with the number of observations, so they generally cannot be applied to very large high-dimensional data sets. Nonlinear mapping algorithms, such as autoencoders, are generally painfully slow to train, and are prone to getting stuck in local minima.

We have recently discovered a way to make nonlinear autoencoders work considerably better ([5]) compared to widely used methods such as PCA, SVD, and LLE. An autoencoder consists of deterministic encoder and decoder networks that convert the high-dimensional data into a low-dimensional code and then decode the code into a reconstruction of the data (Fig. 1). The standard way to train autoencoders is to use backpropagation to reduce the reconstruction error.

It is difficult, however, to optimize the weights in non-linear autoencoders that have multiple hidden layers with many million parameters [2, 3]. With large initial weights, autoencoders typically find poor local minima and with small initial weights, the gradients in the early layers are tiny, making it infeasible to train autoencoders with many hidden layers. This is the main reason why this potentially powerful nonlinear dimensionality reduction algorithm has not found applications in practice. If the initial weights are close to a good solution gradient descent works well, but finding such initial weights requires a very different type of algorithm that learns one layer of features at a time.

Hinton et al.[4] describe a way to perform fast, greedy learning of deep, directed belief networks by training a stack of Restricted Boltzmann Machines one layer at a time. Similarly, we use this greedy learning algorithm to pretrain autoencoders. This learning is treated as *pretraining* stage that progressively discovers low-dimensional representations.

After pretraining stage, the model is "unfolded" as shown in Fig. 1 to produce encoder and decoder networks that initially use the same weights. The global fine-tuning stage then uses backpropagation through the whole autoencoder to fine-tune the weights for optimal reconstruction. The key idea is that this greedy learning algorithm will perform a global search for a good, sensible region in the parameter space. Therefore, with this pretraining, we will already have a good data reconstruction model. Backpropagation is better at local fine-tuning of the model parameters than global search. So further training of the entire autoencoder using backpropagation will result in a good local optimum.

We conducted a set of experiments across various datasets to determine whether our algorithm is capable of discovering low-dimensional codes from high-dimensional data that work much better than linear methods such as PCA, SVD as well as non-parametric methods such as LLE [6].

We first trained a very deep autoencoder on a synthetic dataset containing images of "curves" that were generated from three randomly chosen points in 2-dimensions. For this dataset, the true intrinsic dimensionality is 6, and the relationship between the pixel intensities and the six numbers used to generate them is highly

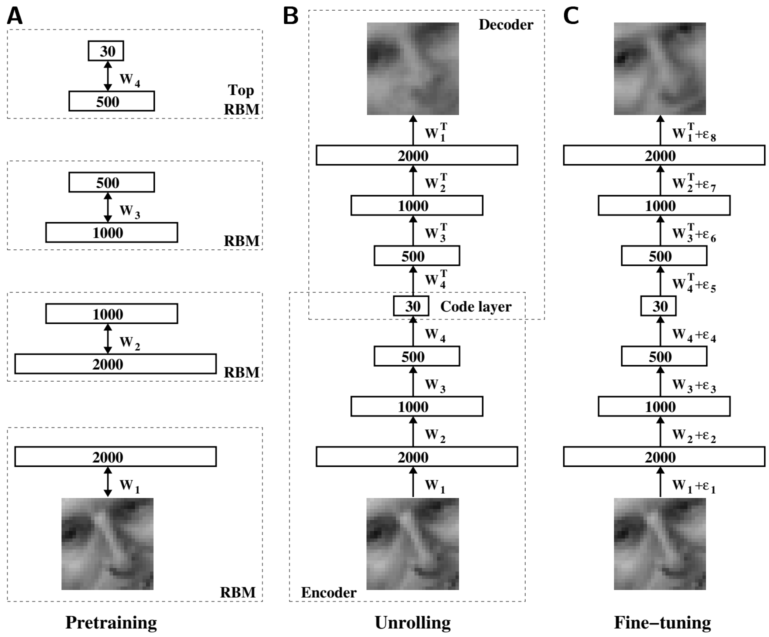

Figure 1: Pretraining consists of learning a stack of Restricted Boltzmann Machines (RBM's) each having only one layer of feature detectors. The learned feature activations of one RBM are used as the "data" for training the next RBM in the stack. After the pretraining, the RBM's are "unrolled" to create a deep autoencoder which is then fine-tuned using backpropagation of error derivatives.

non-linear. The autoencoder consisted of an encoder with layers of size (28x28)-400-200-100-50-25-6 and a symmetric decoder. Figure 2 shows that the autoencoder discovered how to convert each 784 pixel image into 6 real numbers that allow almost perfect reconstruction. PCA gives much worse reconstructions.

Next, we used a 784-1000-500-250-30 autoencoder to extract codes for all the handwritten digits in the MNIST training set[1] After fine-tuning on all 60,000 training images, the autoencoder was tested on 10,000 new images and produced much better reconstructions than PCA (see Fig. 2). We also used a 625-2000-1000-500-30 autoencoder to discover 30-dimensional codes for grey-level image patches that were derived from the Olivetti face dataset[2]. Figure 2 shows that the autoencoder significantly outperforms PCA.

When applied in the domain of information retrieval, autoencoders can be trained to extract low-dimensional semantic codes from a large collection of documents. Learned low-dimensional codes can then be used for fast and accurate document retrieval. We trained a 2000-500-250-125-10 autoencoder on half of Reuters 804,414 newswire stories, whereas the other half was used for testing. Each story was represented as a vector of document-specific probabilities of the 2000 commonest word stems[3]. Figure 3 shows that the autoencoder significantly outperforms Latent Semantic Analysis [1], a well-known document retrieval method based on PCA. Autoencoders also outperform LLE [5].

Unlike non-parametric methods such as LLE, ISOMAP [6, 7], autoencoders give mappings in both directions between the data and code spaces and they can be applied to very large datasets because both the pretraining and the fine-tuning scale linearly in time and space with the number of training cases. This potentially makes autoencoders one of the best nonlinear dimensionality reduction tools for large data sets.

(I would prefer a short contributed talk. If time permits I will also briefly describe how this methodology can be applied to the task of learning a nonlinear embedding).

References

[1] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[2] D. DeMers and G.W. Cottrell. Nonlinear dimensionality reduction. In Cowan Hanson and Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 580–587, San Mateo, CA, 1993. Morgan Kaufmann.

[1] The MNIST dataset is available at http://yann.lecun.com/exdb/mnist/index.html

[2] The Olivetti face dataset is available at http://www.cs.toronto.edu/ roweis/data.ht ml

[3] The Reuter Corpus Volume 2 is available at http://trec.nist.gov/data/reuters/reuters.html

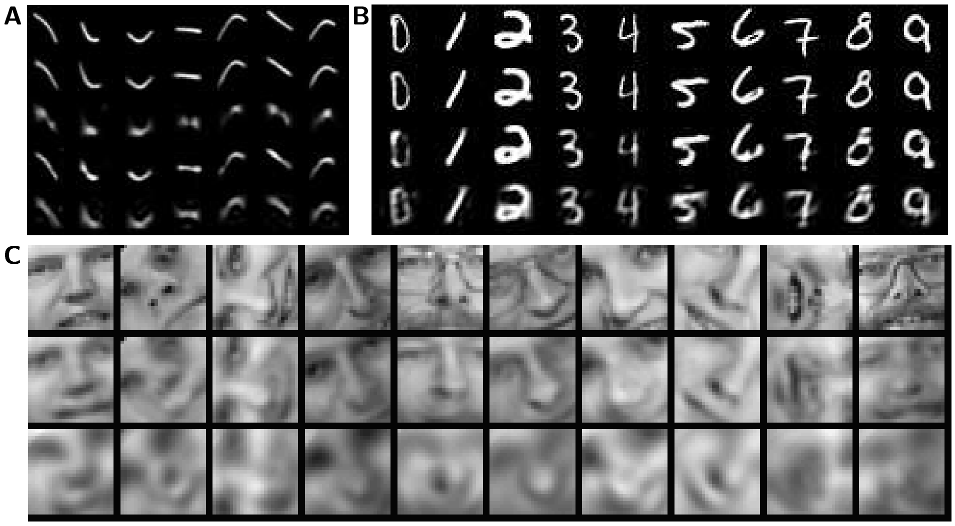

Figure 2: Top left panel (by row): Random samples of curves from the test dataset; reconstructions produced by the 6-dimensional deep autoencoder; reconstructions by "logistic PCA" using 6 components; reconstructions by logistic and standard PCA using 18 components. The average squared error per image for the last four rows is 1.44, 7.64, 2.45, 5.90. Top right panel (by row): A random test image from each class; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional logistic PCA and standard PCA. The average squared errors for the last three rows are 3.00, 8.01, and 13.87. Bottom panel (by row): Random samples from the test dataset; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional PCA. The average squared errors are 126 and 135.

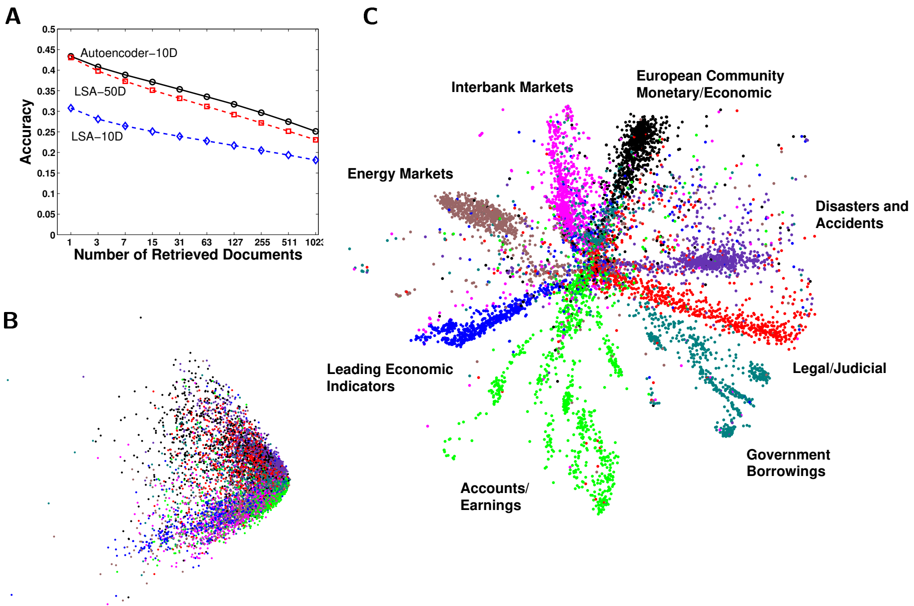

Figure 3: Top left: The fraction of retrieved documents in the same class as the query when a query document from the test set is used to retrieve other test set documents, averaged over all 402,207 possible queries. Bottom left: The codes produced by 2-dimensional LSA. Right: The codes produced by a 2000-500-250-125-2 autoencoder.

[3] R. Hecht-Nielsen. Replicator neural networks for universal optimal source coding. *Science*, 269:1860–1863, 1995.

[4] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 2006.

[5] G. E. Hinton and R. R. Salakhutdinov. Non-linear dimensionality reduction using neural networks. *To appear in Science*, 2006.

[6] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[7] J. A. Tenenbaum, V. J. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.