Machine Learning

Lecture 19: Even more on neural networks

Stefan Harmeling

18. December 2017

Unsupervised neural networks

Auto-associative neural network (autoencoder)

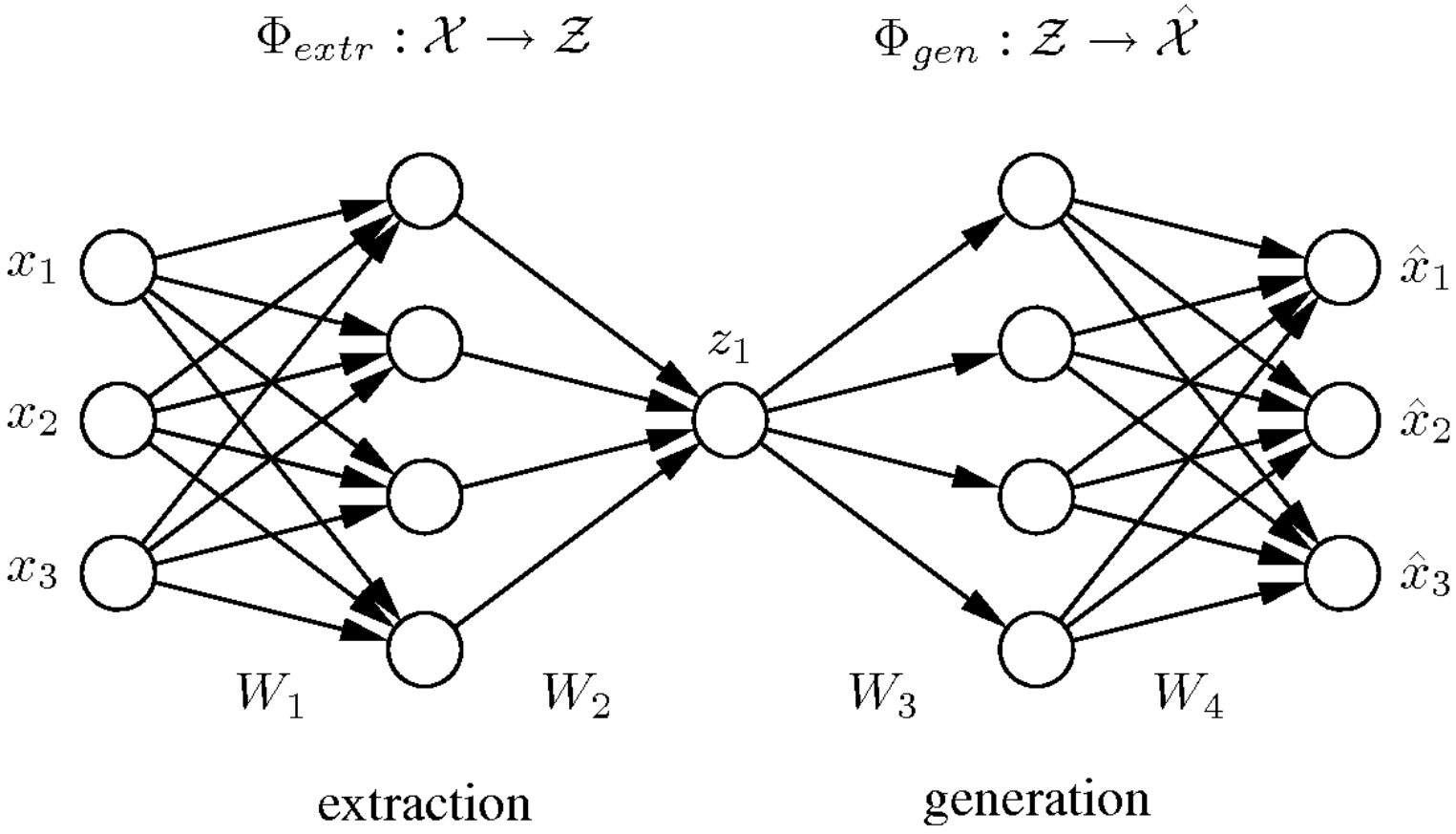
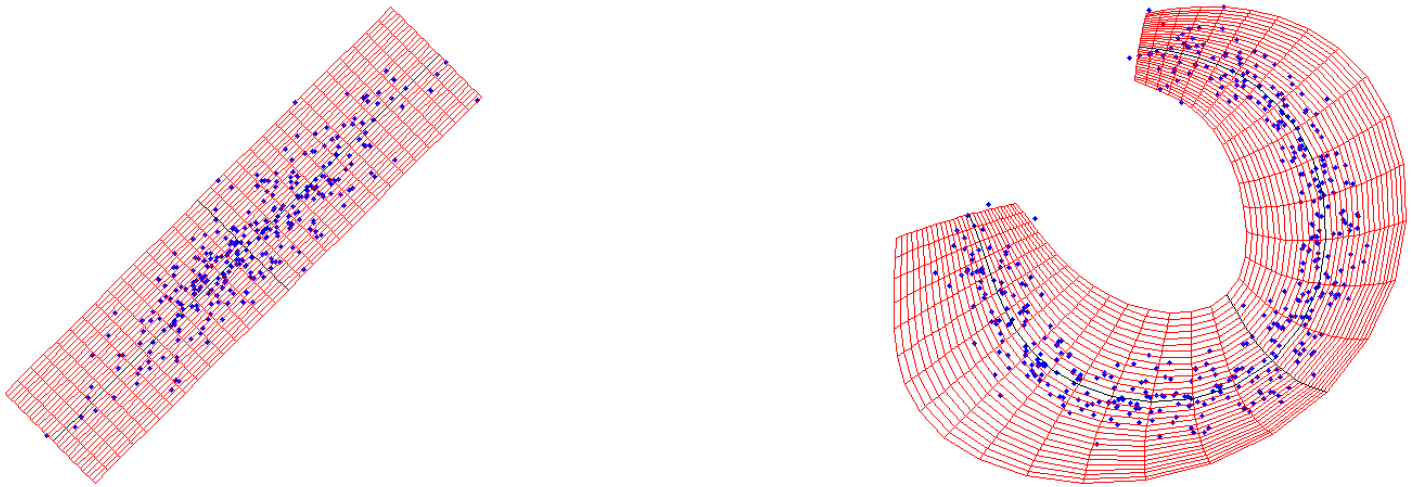


image from Matthias Scholz, `http://nlpca.org` and Scholz, *Nichtlinear Hauptkomponentenanalyse auf Basis neuronaler Netze*, Diplomarbeit, 2002.

see also, Kramer, *Nonlinear principal component analysis using auto- associative neural networks*, AIChE Journal, 1991.

Auto-associative neural network (autoencoder)



from Matthias Scholz, `http://nlpca.org`

Auto-associative neural network (autoencoder)

What is optimized?

- The neural network can be written as follows:

$$f(x) = b_4 + W_4 \tanh(b_3 + w_3 \tanh(b_2 + w_2^T \tanh(b_1 + W_1 x)))$$

  where W_4, W_1 are matrices, w_3, w_2, b_4, b_3, b_1, x are vectors, and b_2 is a scalar.
- the loss function to minimize is

$$E = \|x - f(x)\|^2$$

  i.e. we are trying to reconstruct the input with the output
- network reconstructs a single nonlinear directions (probably of largest variance)

Auto-associative neural network (autoencoder)

What if we want to reconstruct a two dimensional embedding?

- The neural network can be written as follows:

$$f(x) = b_4 + W_4 \tanh(b_3 + W_3 \tanh(b_2 + W_2^T \tanh(b_1 + W_1 x)))$$

  where W_4, W_3, W_2, W_1 are matrices, b_4, b_3, b_2, b_1, x are vectors.
- the loss function to minimize is

$$E = \|x - f(x)\|^2$$

- note that if there are several dimensions for the bottleneck there is no order on the nodes (unlike linear PCA), i.e. it is only about the reconstruction error (same problem as on slide 24 in lecture 12, where we found the two dimensional subspace of largest variance, but with an arbitrary rotation on the single directions),
- Scholz (2002) calls this s-NLPCA, “s” like symmetric.

Auto-associative neural network (autoencoder)

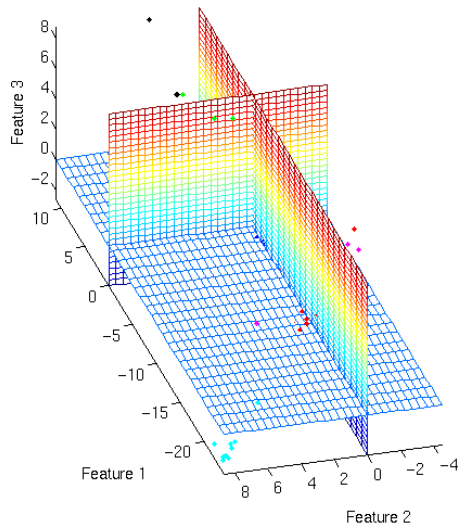
(Hierarchical) h-NLPCA, Scholz, 2002

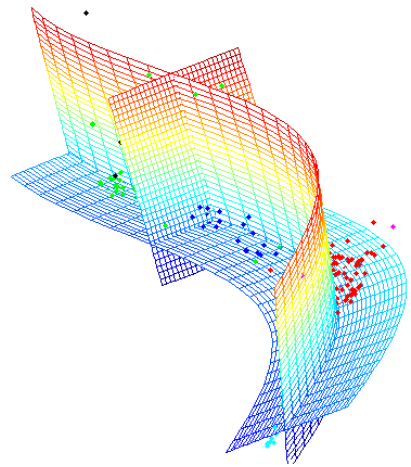
- Scholz suggests an *hierarchical* error function (called h-NLPCA, "h" like hierarchical, e.g. for two dimensions):

$$E = \alpha E_1 + E_{12} \qquad \text{for } 0 \leq \alpha < \infty$$

- For $\alpha = \infty$ we get one-dimensional NLPCA, for $\alpha = 0$ we get s-NLPCA, in between he calls it h-NLPCA.

Auto-associative neural network (autoencoder)

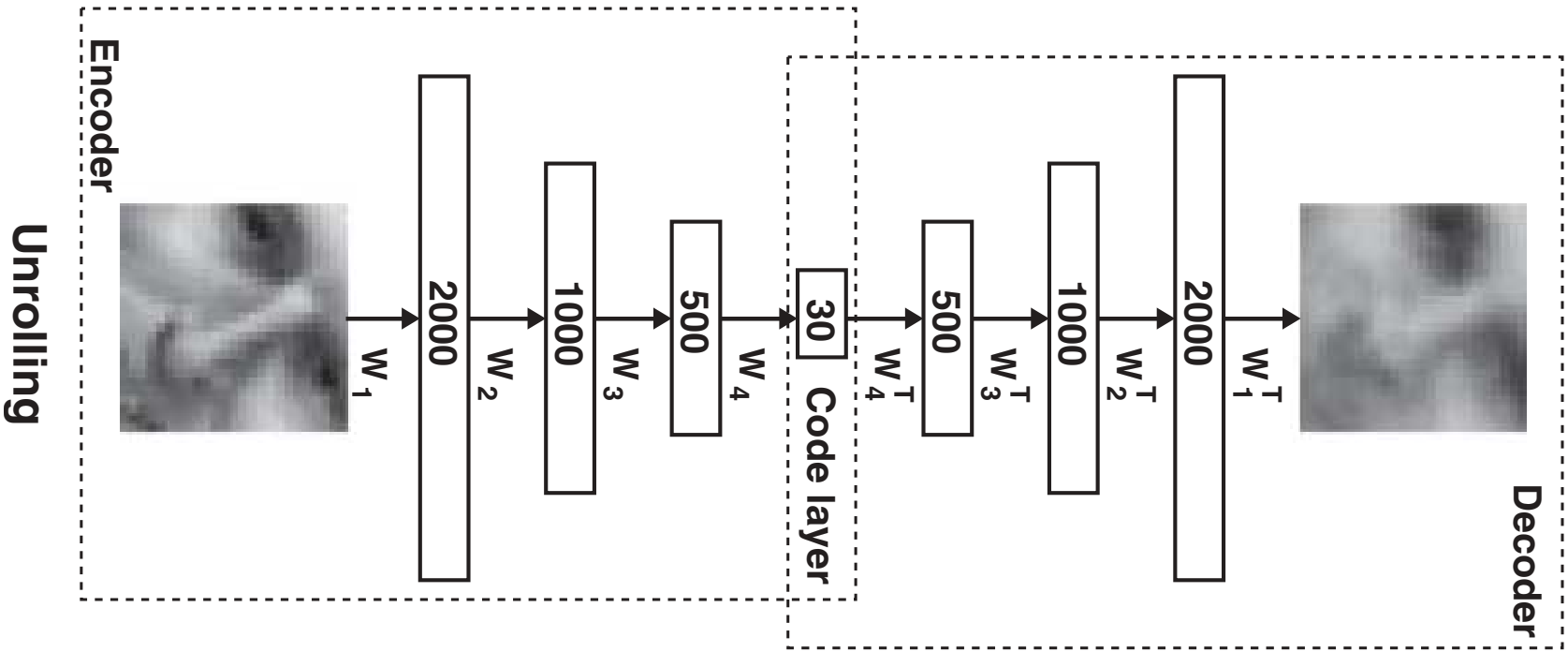




from Matthias Scholz, `http://nlpca.org`

Neural networks with pretraining

A recent unsupervised network, Hinton/Salahutdinov, Science, 2006



from Hinton and Salakhutdinov, *Reducing the Dimensionality of Data with Neural Networks*, Science, 2006.

Neural networks with pretraining

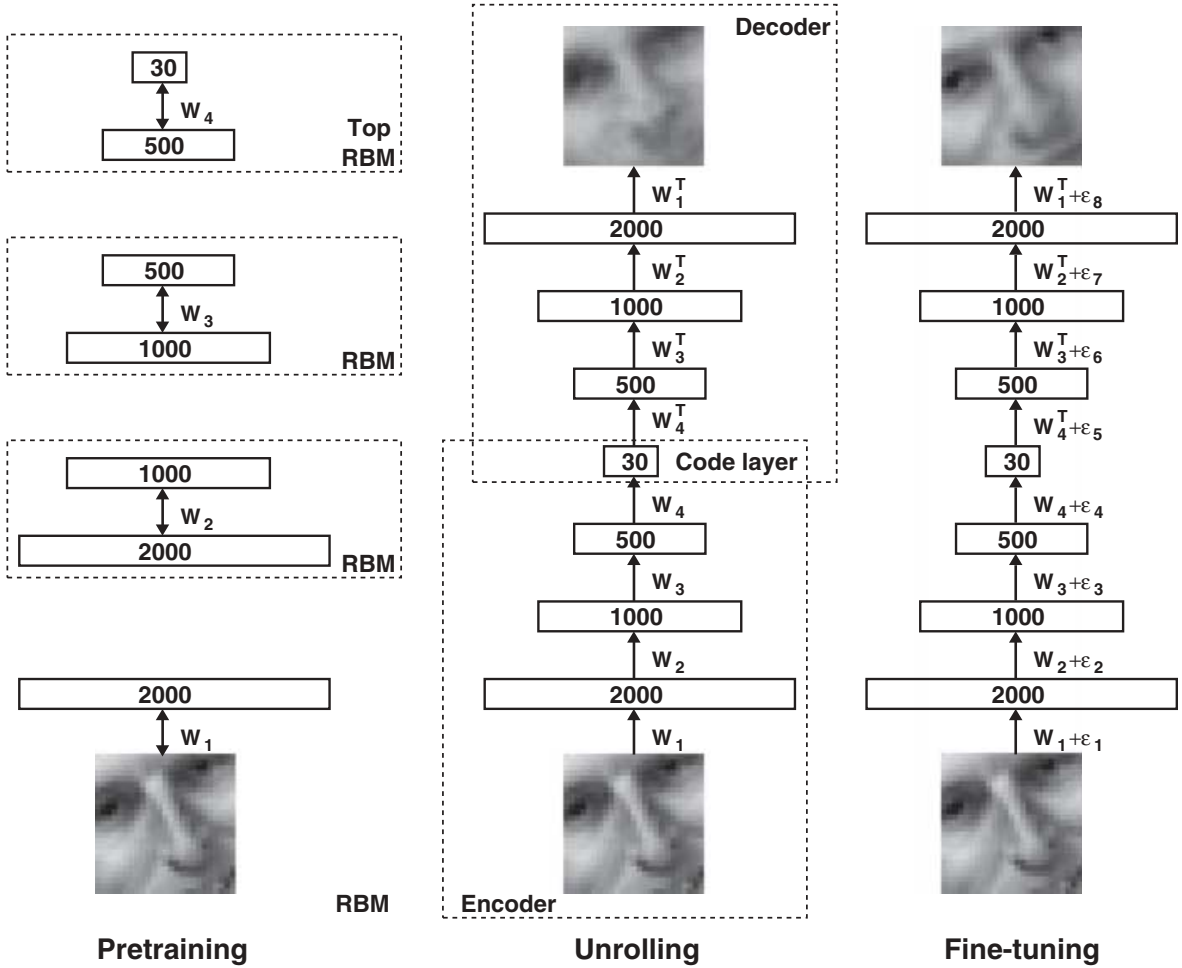


Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the "data" for training the next RBM in the stack. After the pretraining, the RBMs are "unrolled" to create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

from Hinton and Salakhutdinov, *Reducing the Dimensionality of Data with Neural Networks*, Science, 2006.

Neural networks with pretraining

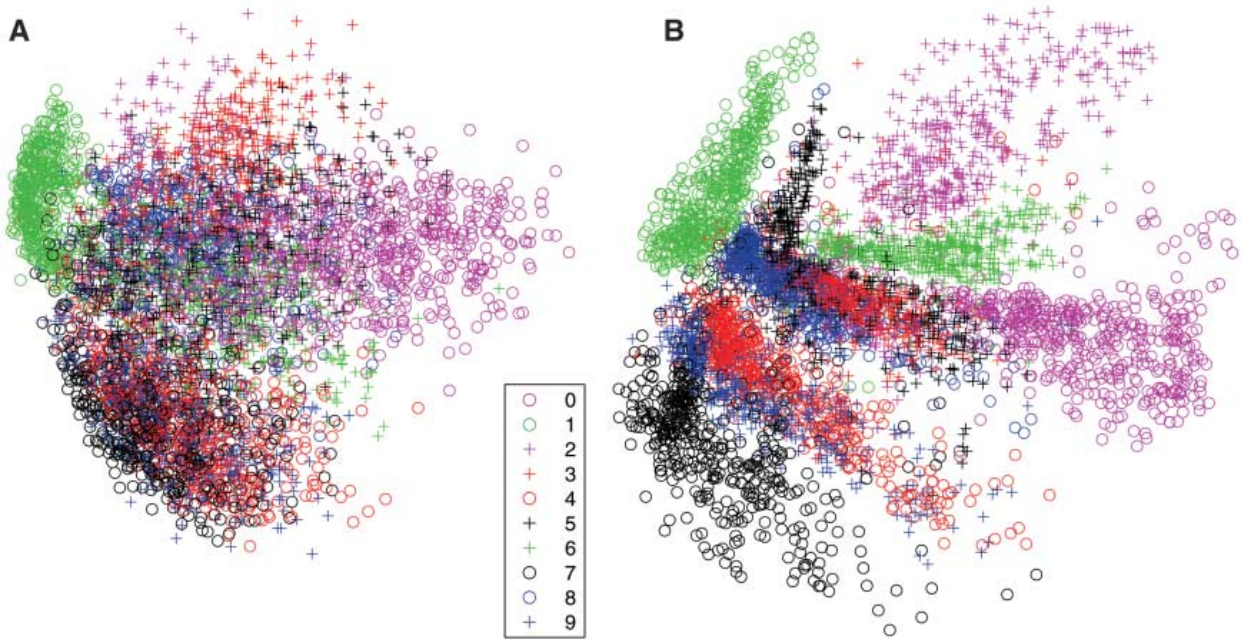


Fig. 3. (**A**) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (**B**) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder. For an alternative visualization, see (*8*).

from

Hinton and Salakhutdinov, *Reducing the Dimensionality of Data with Neural Networks*, Science, 2006.

Installing Tensorflow

`https://github.com/tensorflow/tensorflow/tree/r1.4`

DevDocs

`http://devdocs.io`

On the web

- Deep learning in your browser with Javascript (Andrej Karpathy) `http://cs.stanford.edu/people/karpathy/convnetjs/`
- Hacker's guide to Neural Networks (Andrey Karpathy) `http://karpathy.github.io/neuralnets/`
- Neural Networks, Manifolds, and Topology (Christopher Olah) `https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/`
- The unreasonable effectiveness of recurrent neural networks `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`
- Understanding LSTM networks (Christopher Olah) `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`

Topics for BSc/MSc thesis

- Dig out an old neural networks paper and reimplement it with modern technology.
- Style transfer with music
- Try non-component-wise activation functions?
- Continue Harry Potter with neural networks (someone did!).
- Design a better interface between python and tensorflow.
- Design a specialized programming language for this.
- Are there neural networks that can run forward and backward (invertible)?
- Your cool idea!

If you are interested in writing your BSc thesis in Machine learning (with me or Julius Ramakers), please write me an email. I will maintain a list.