

# Neural Networks and Deep Learning

## Unsupervised, Representation, and Deep Learning

Nicholas Dronen

Department of Computer Science  
[dronen@colorado.edu](mailto:dronen@colorado.edu)

February 25, 2019



University of Colorado **Boulder**

Autoencoders

Undercomplete Autoencoders

Regularized Autoencoders

    Sparse Autoencoders

    Denoising Autoencoders

    Contractive Autoencoders

Representational Power, Layer Size  
and Depth

Learning Manifolds with  
Autoencoders

Applications of Autoencoders

Representation Learning

    Greedy Layer-Wise

    Unsupervised Pretraining

    Semi-Supervised Disentangling  
of Causal Factors

    Distributed Representations

    Exponential Gains from Depth

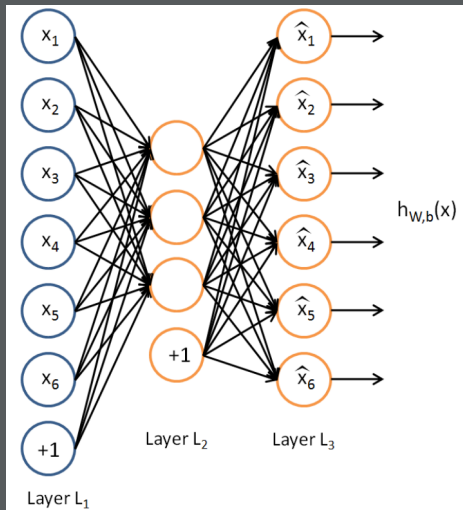
    Providing Clues to Discover

    Underlying Causes



## Autoencoders

- A typical autoencoder maps input  $\mathbf{x}$  to an output reconstruction  $\hat{\mathbf{x}}$  through an internal representation  $\mathbf{h}$ .
- The encoder,  $f$ , learns the stochastic mapping  $p_{encoder}(\mathbf{h}|\mathbf{x})$  and the decoder,  $g$ , learns  $p_{decoder}(\mathbf{x}|\mathbf{h})$



Source: Ng, 2011

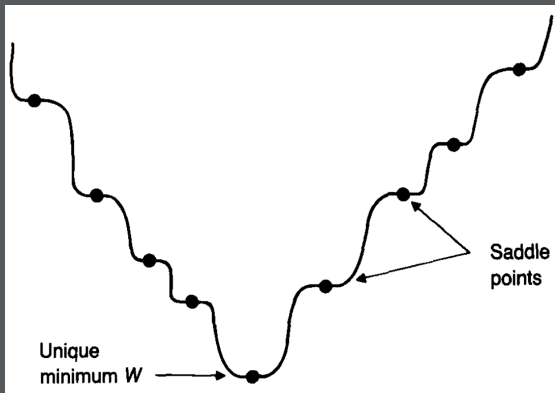


- In an *undercomplete* autoencoder, the code  $\mathbf{h}$  has lower dimensionality than the input  $\mathbf{x}$ .
- The loss function  $L(\mathbf{x}, g(f(\mathbf{x})))$  penalizes  $g(f(\mathbf{x}))$  for being different from  $\mathbf{x}$
- In this scenario,  $\mathbf{h}$  is constrained and the encoder must learn the essential features of the data.



## Linear Autoencoders

When encoder and decoder are linear and  $L$  is mean squared error, an undercomplete autoencoder learns to span the same *subspace* as PCA. How do PCA and a linear autoencoder differ?



Source: Baldi and Hornik, 1989



## Overcomplete Autoencoders

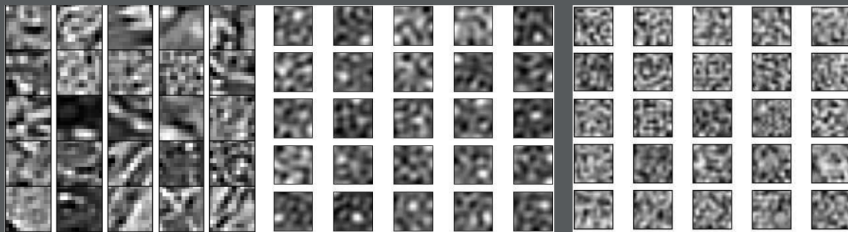
- In an *overcomplete* autoencoder, the code  $\mathbf{h}$  has greater dimensionality than the input  $\mathbf{x}$ .
- Overcompleteness can cause overfitting. How?
- The (compound) loss function of a regularized autoencoder contains a term that encourages the model to learn a mapping from inputs to outputs that is richer than the identity, via:
  - » Sparse representation
  - » Robustness to noise
  - » Minimization of the encoder gradient with respect to the input.



Figure 1: Regularization forces the auto-encoder to become less sensitive to the input, but minimizing reconstruction error forces it to remain sensitive to variations along the manifold of high density. Hence the representation and reconstruction end up capturing well variations on the manifold while mostly ignoring variations orthogonal to it.

Source: [Alain and Bengio, 2013](#)





Source: [Vincent et al, 2010](#) Regular autoencoder trained on natural image patches. Left: some of the  $12 \times 12$  image patches used for training. Middle: filters learnt by a regular under-complete autoencoder (50 hidden units) using tied weights and L2 reconstruction error. Right: filters learnt by a regular over-complete autoencoder (200 hidden units). The under-complete autoencoder appears to learn rather uninteresting local blob detectors. Filters obtained in the overcomplete case have no recognizable structure, looking entirely random.

## Sparse Autoencoders

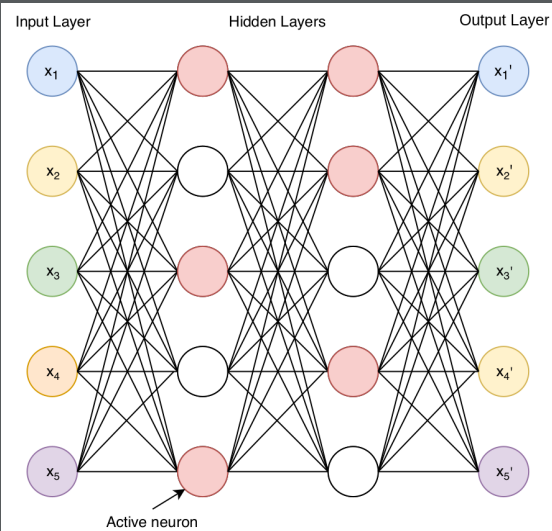
- The training of a sparse autoencoder involves a sparsity penalty  $\Omega(h)$  on code layer  $\mathbf{h}$ , in addition to reconstruction layer.

$$L(x, g(f(x))) + \Omega(\mathbf{h})$$

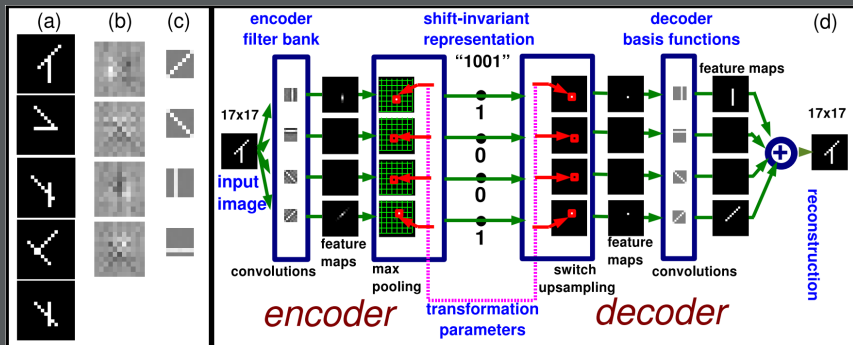
- Advantages of sparsity (Source: [Glorot, 2011](#))
  - » Information disentangling.
  - » Efficient variable-size representation.
  - » Linear separability.
  - » Distributed but sparse.







# Sparse Autoencoder



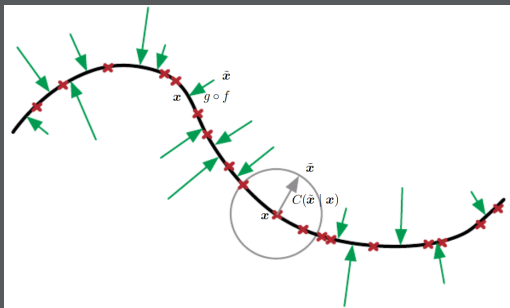
Left Panel: (a) sample images from the “two bars” dataset. Each sample contains two intersecting segments at random orientations and random positions. (b) Non-invariant features learned by an auto-encoder with 4 hidden units. (c) Shift-invariant decoder filters learned by the proposed algorithm. The algorithm finds the most natural solution to the problem. Right Panel (d): architecture of the shift-invariant unsupervised feature extractor applied to the two bars dataset. The encoder convolves the input image with a filter bank and computes the max across each feature map to produce the invariant representation. The decoder produces a

## Denoising Autoencoders

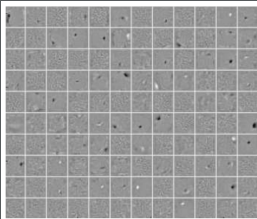
A denoising autoencoder minimizes  $L(x, g(f(\hat{x})))$ , where  $\hat{x}$  is a copy of  $x$  that has been corrupted by some form of noise.



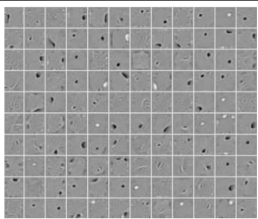
Image from MNIST dataset corrupted by masking noise



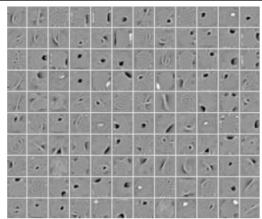
A denoising autoencoder is trained to map a corrupted data point  $\hat{x}$  to the original data point  $x$



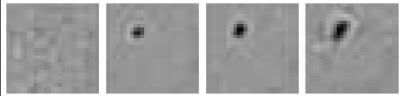
(a) No corruption



(b) 25% corruption



(c) 50% corruption

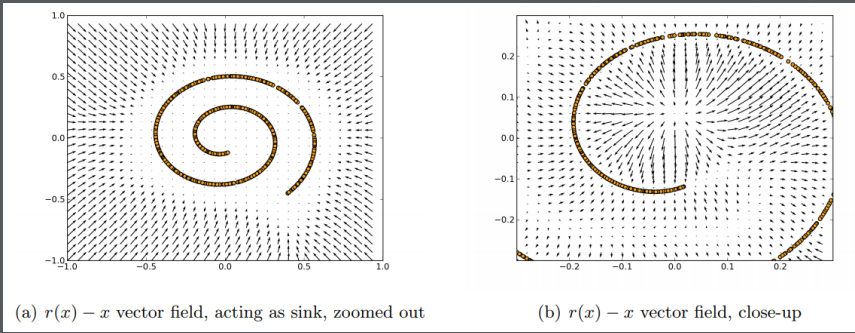


(d) Neuron A (0%, 10%, 20%, 50% corruption)

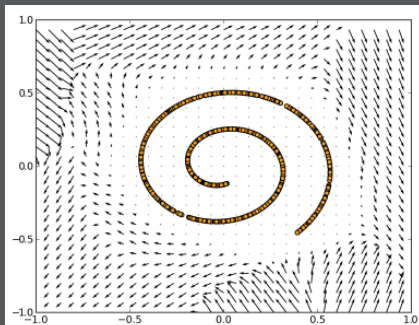


(e) Neuron B (0%, 10%, 20%, 50% corruption)

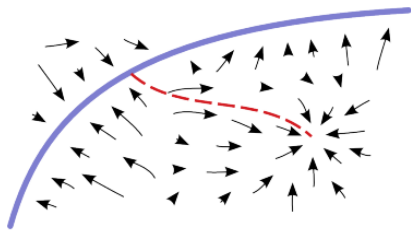
Source: [Vincent et al, 2010](#)



Source: [Alain and Bengio, 2013](#)



(a) DAE misbehaving when away from manifold







(b) sampling getting trapped into bad attractor

Source: Alain and Bengio, 2013



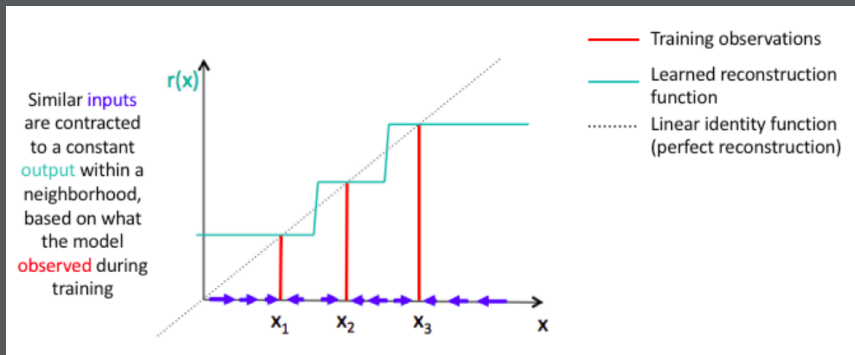
The regularization term of the contractive autoencoder minimizes the gradient of  $f$  with respect to  $\mathbf{x}$ .

$$\Omega(\mathbf{h}) = \lambda \left\| \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right\|_F^2$$

Input point	Tangent vectors
	
	
	Contractive autoencoder

Source: <http://www.deeplearningbook.org>





Source: [Alain and Bengio, 2013](#)



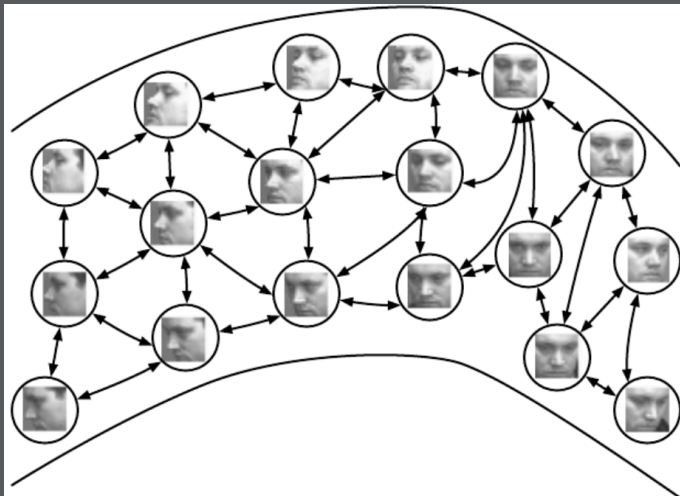


Experimentally, deep autoencoders yield much better compression than corresponding shallow or linear autoencoders.



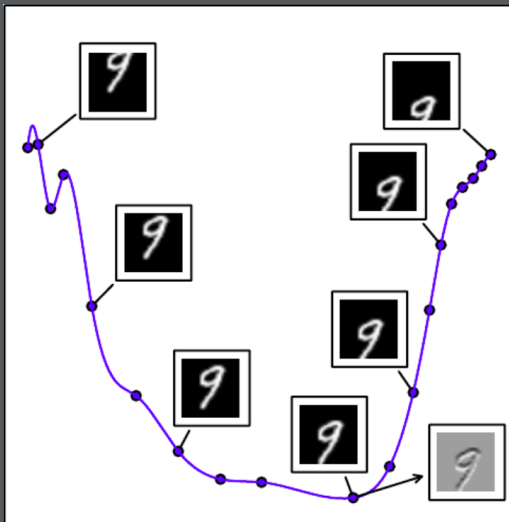
Source: [Hinton and Salakhutdinov, 2006](#)

1. A random test image from each class.
2. Reconstructions by a 30-dimensional autoencoder.
3. Reconstructions by 30-dimensional logistic PCA.
4. Reconstructions by 30-dimensional standard PCA



Nonparametric manifold learning based on nearest neighbor graph.

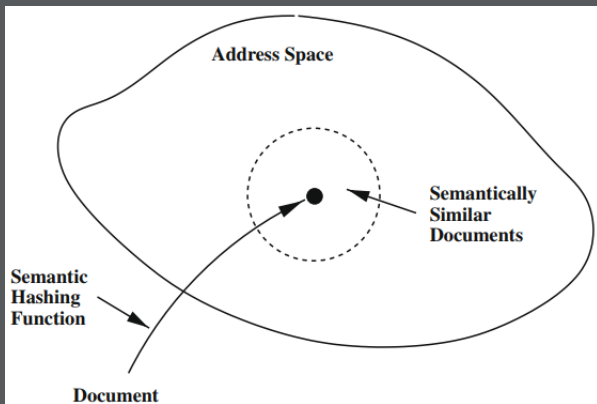
Source: <http://www.deeplearningbook.org>



1-D manifold of an MNIST digit translated vertically.

Source: <http://www.deeplearningbook.org>

## Semantic Hashing

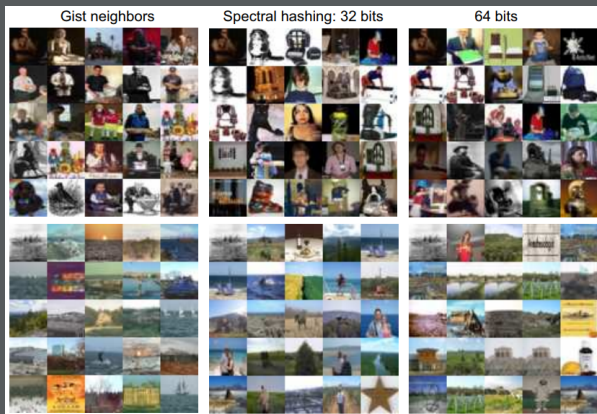


A schematic representation of semantic hashing.

Source: [Salakhutdinov and Hinton, 2008](#)



## Spectral Hashing



Retrieval results on a dataset of 80 million images using the original gist descriptor, and hash codes build with spectral hashing with 32 bits and 64 bits. Source: [Torralba et al, 2008](#)

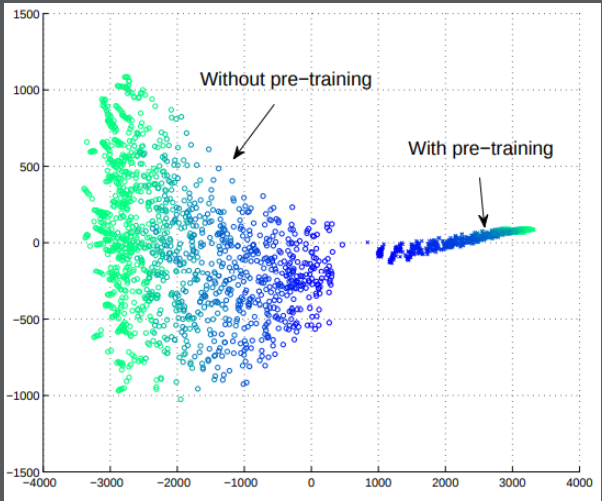


## Greedy Layerwise Unsupervised Pretraining - (GLUP)

- Greedy - it optimizes each piece of the solution independently, one piece at a time.
- Layerwise - the independent pieces are the layers of the network, the  $k$ -th layer is trained while keeping the previous ones fixed.
- Unsupervised - each layer is trained with an unsupervised representation learning algorithm.
- Pretraining - it is supposed to be only a first step before a joint training algorithm is applied to fine-tune all the layers together.

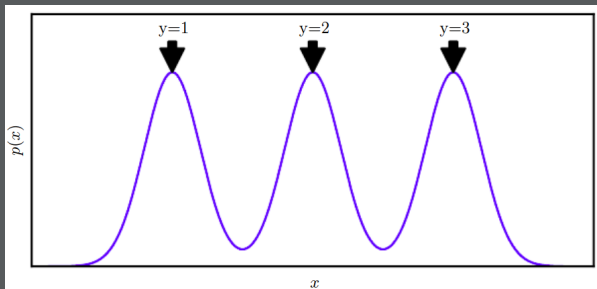
This is not commonly done these days. Why not?





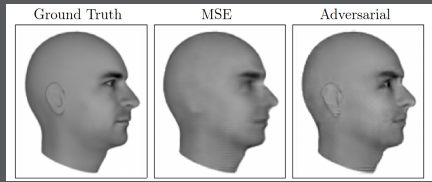
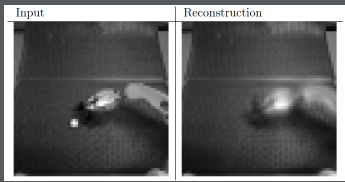
Source: Erhan et al, 2010

A good representation is one that reveals the underlying causal factors of the data. If finding  $p(\mathbf{x})$  makes finding  $p(y|\mathbf{x})$  easier, then semi-supervised learning can help.



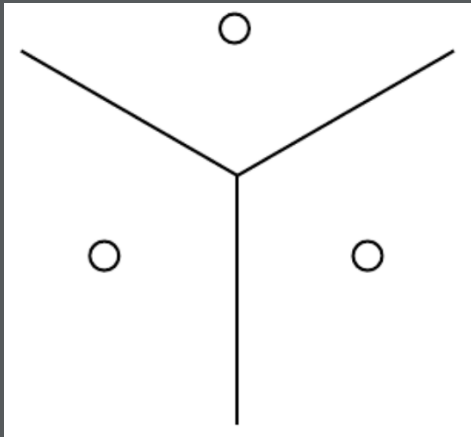
Source: <http://www.deeplearningbook.org>





Source: <http://www.deeplearningbook.org>

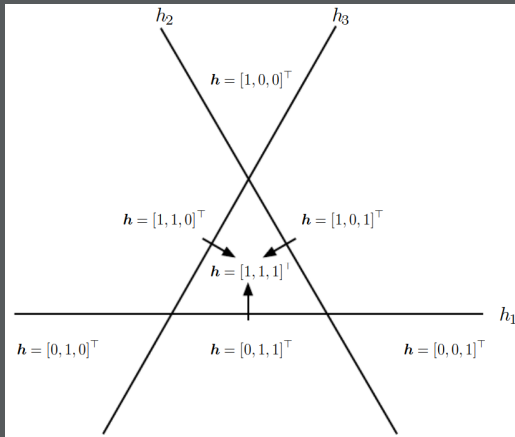
## Non-Distributed Representations



With non-distributed representations, the data can only be encoded naively. Decision trees and  $k$ -nearest neighbors work this way.

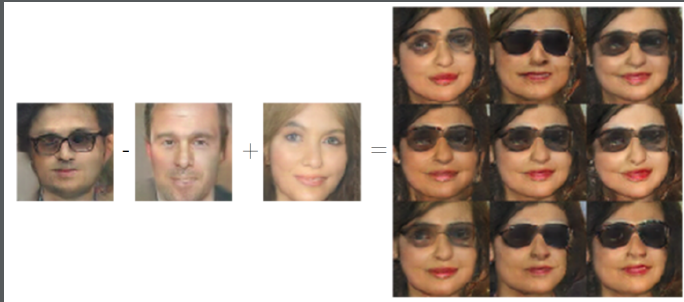


## Distributed Representations

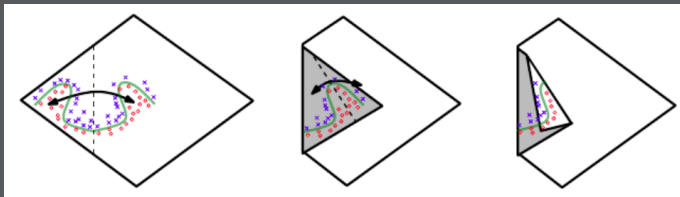


Distributed representations partition/encode the data such that similarity structure naturally emerges.





Source: <http://www.deeplearningbook.org>



An intuitive, geometric explanation of the exponential advantage of deeper rectifier networks.  
Source: [Montufar et al, 2014](#)

- Deep models encode a very general belief that the function we want to learn should involve composition of several simpler functions.
- Empirically, greater depth does seem to result in better generalization for a wide variety of tasks.



## Generic Regularization strategies

- Smoothness
- Linearity
- Multiple explanatory factors
- Causal factors
- Depth, or a hierarchical organization of explanatory factors
- Shared factors across tasks
- Manifolds
- Natural clustering
- Temporal and spatial coherence
- Sparsity
- Simplicity of factor dependencies

