

Learning Manifolds with Autoencoders

Sargur Srihari
srihari@buffalo.edu

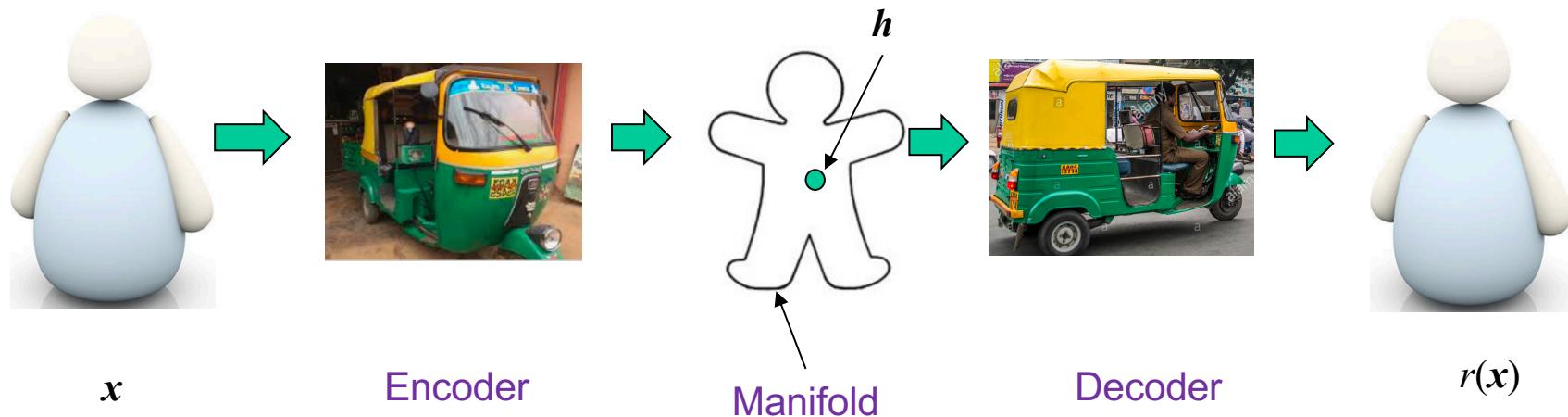
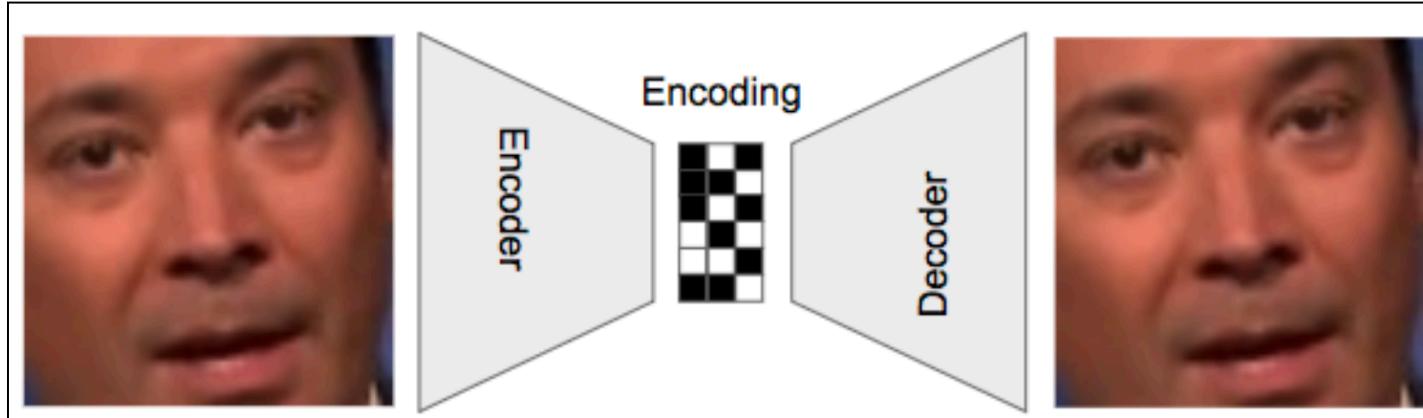
Topics in Autoencoders

- What is an autoencoder?
 1. Undercomplete Autoencoders
 2. Regularized Autoencoders
 3. Representational Power, Layout Size and Depth
 4. Stochastic Encoders and Decoders
 5. Denoising Autoencoders
 6. Learning Manifolds with Autoencoders
 7. Contractive Autoencoders
 8. Predictive Sparse Decomposition
 9. Applications of Autoencoders

Topics in Learning Manifolds with Autoencoders

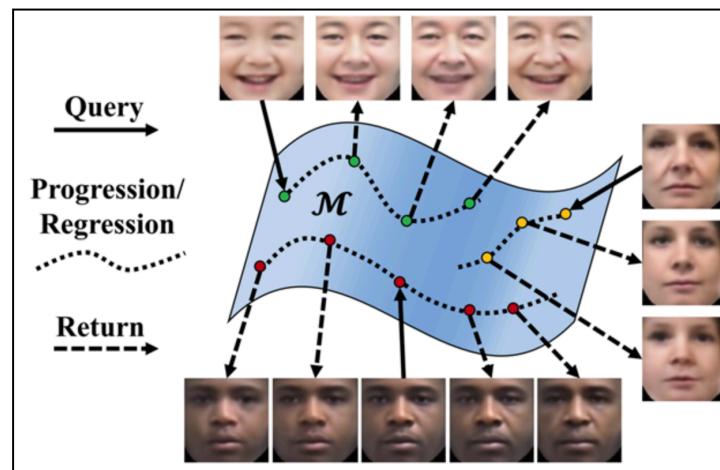
- Manifold Hypothesis
- Definition of a mathematical manifold
- Manifold in Machine Learning
- Specifying manifolds using tangent planes
- Specialized autoencoders

Autoencoders



Manifold Hypothesis

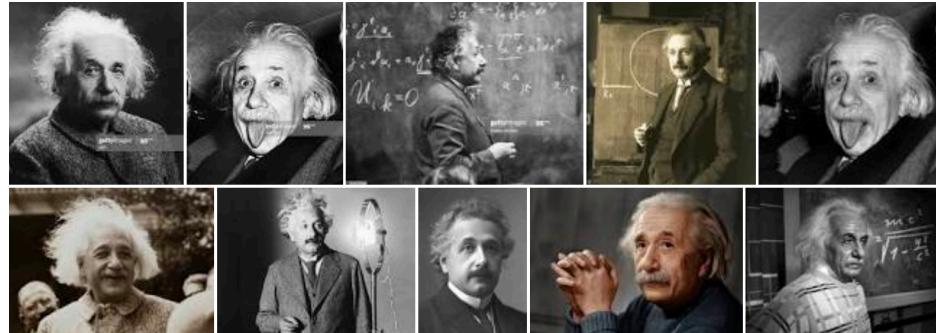
- Data concentrates around a low-dimensional manifold
 - Manifold Hypothesis



- Why study nature of manifolds?
 - Some ML algorithms have unusual behavior if given an input that is off of the manifold
- Autoencoders take this idea further and aim to learn the structure of the manifold

Why does data lie on a Manifold?

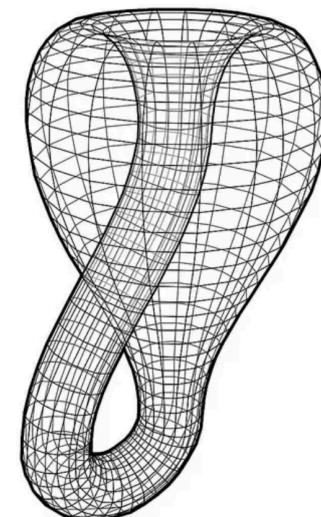
- Suppose we want to classify all (b&w) images with $m \times n$ pixels
 - Each pixel has a numerical value
 - An image is a single point of dimension $N = mn$
- Suppose all $m \times n$ images are photos of Einstein



- We are restricted on choice of values for the pixels
- Random choices will not generate such images
- Therefore, we expect there to be less freedom of choice
- Manifold hypothesis states that this subset should actually live in an (ambient) space of lower dimension, in fact a dimension much, much smaller than N

Reason for Low-dimensional manifolds

- Low dimensional structure arises due to constraints arising from physical laws
- Empirical study
 - Large no. of 3×3 images represented as points in \mathbb{R}^9
 - Lie on a 2-D manifold known as the Klein bottle

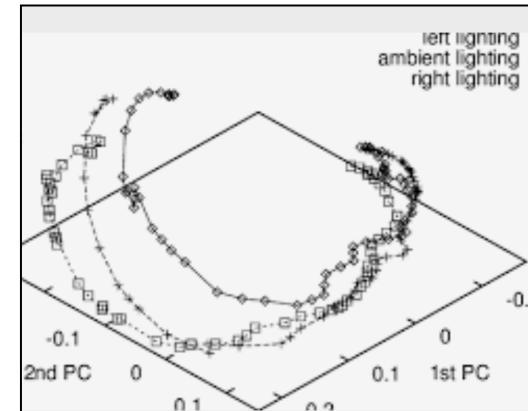


Low-dimensional manifolds embedded in high dimensional spaces

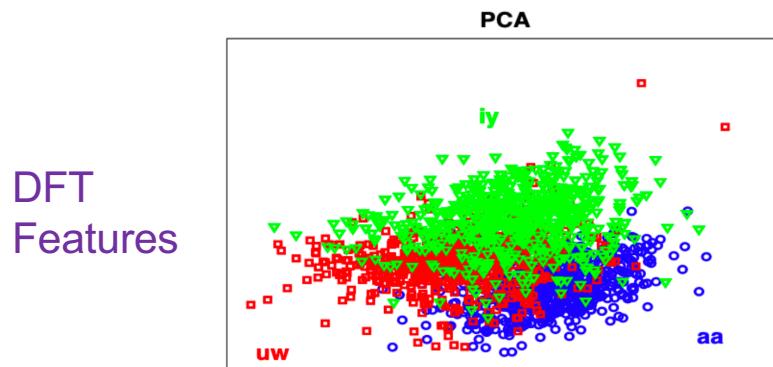
- Image vectors of 3D objects under illuminations, camera views



Manifold formed by three face sequences under different lighting conditions rotating from profile-to-profile (-90° to $+90^\circ$).

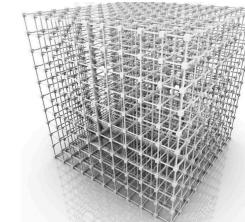


- Phonemes in speech signals

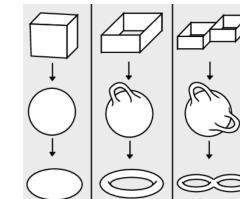
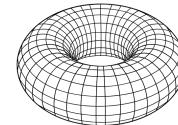


Definition of Manifold

- A *Manifold* is a topological space that locally resembles Euclidean space near each point
 - An n -dimensional manifold is a topological space M for which every point $x \in M$ has a neighborhood *homeomorphic* to Euclidean space R^n

3-D manifold M  R^3 

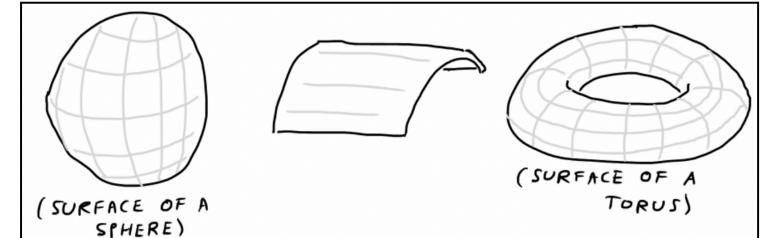
- *Homeomorphism* in topology is also called a continuous transformation
 - One-to-one correspondence in two geometric figures or topological spaces that is continuous in both direction



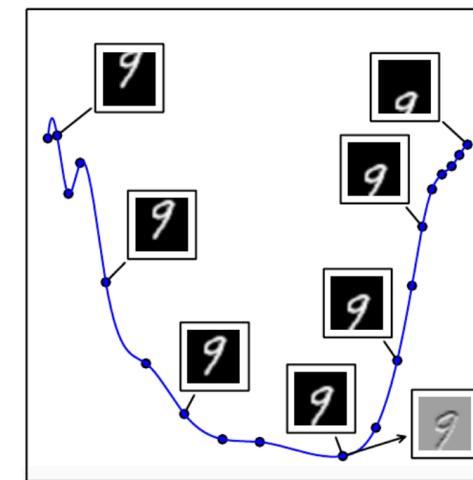
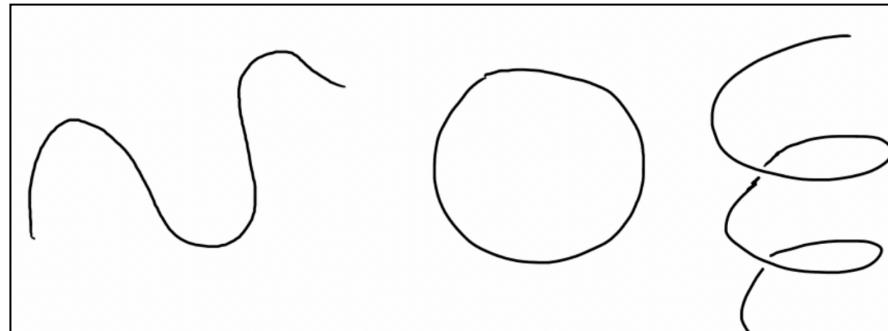
- *Homomorphism* in algebra
 - The most important functions between two groups are those that “preserve” group operations, and they are called homomorphisms
 - A function $f: G \rightarrow H$ between two groups is a homomorphism when $f(xy) = f(x)f(y)$ for all x and y in G

A manifold has a dimension

- A 2-D manifold is a surface

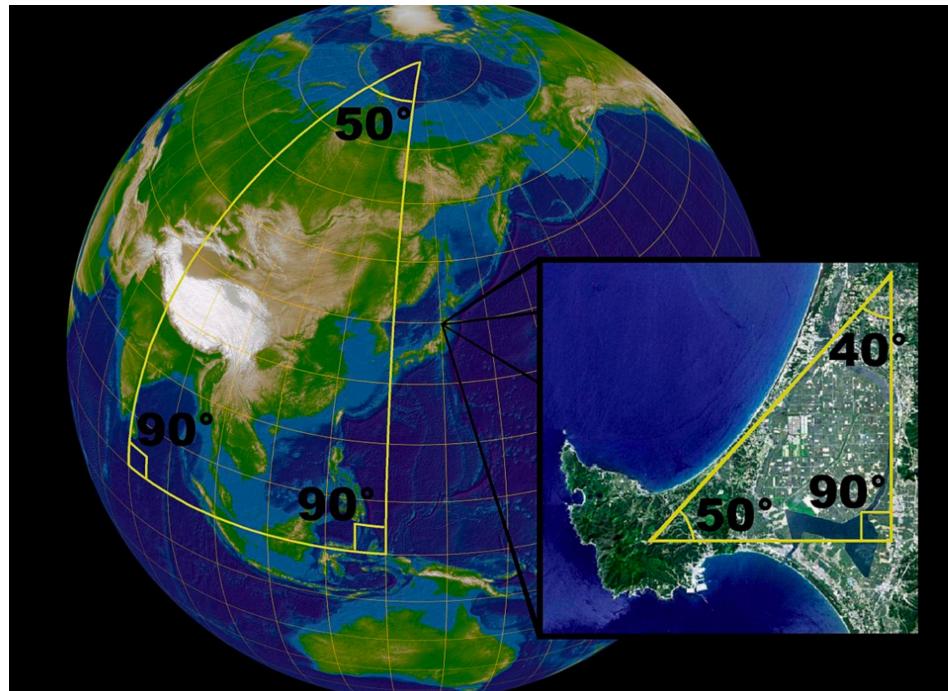


- It could also be a union of several surfaces, too
 - We assume manifolds are connected
- A 1-D manifold is a curve



- A 0-D manifold is a point
- All of 3-space, \mathbb{R}^3 , is a 3-D manifold

2-D Manifold in R^3 homeomorphic to R^2



In mathematics, a manifold is a topological space that locally resembles Euclidean space near each point

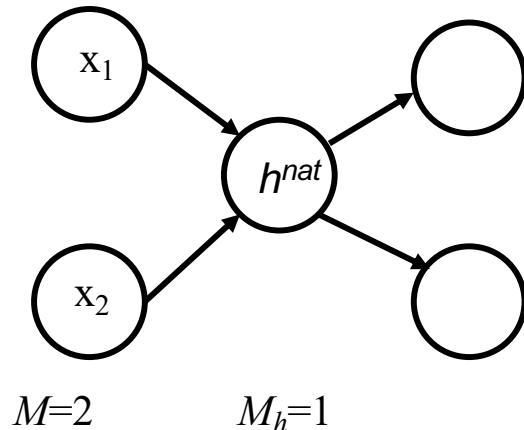
A topological space may be defined as a set of points, along with a set of neighborhoods for each point, satisfying a set of axioms relating points and neighborhoods

Manifold in Machine Learning

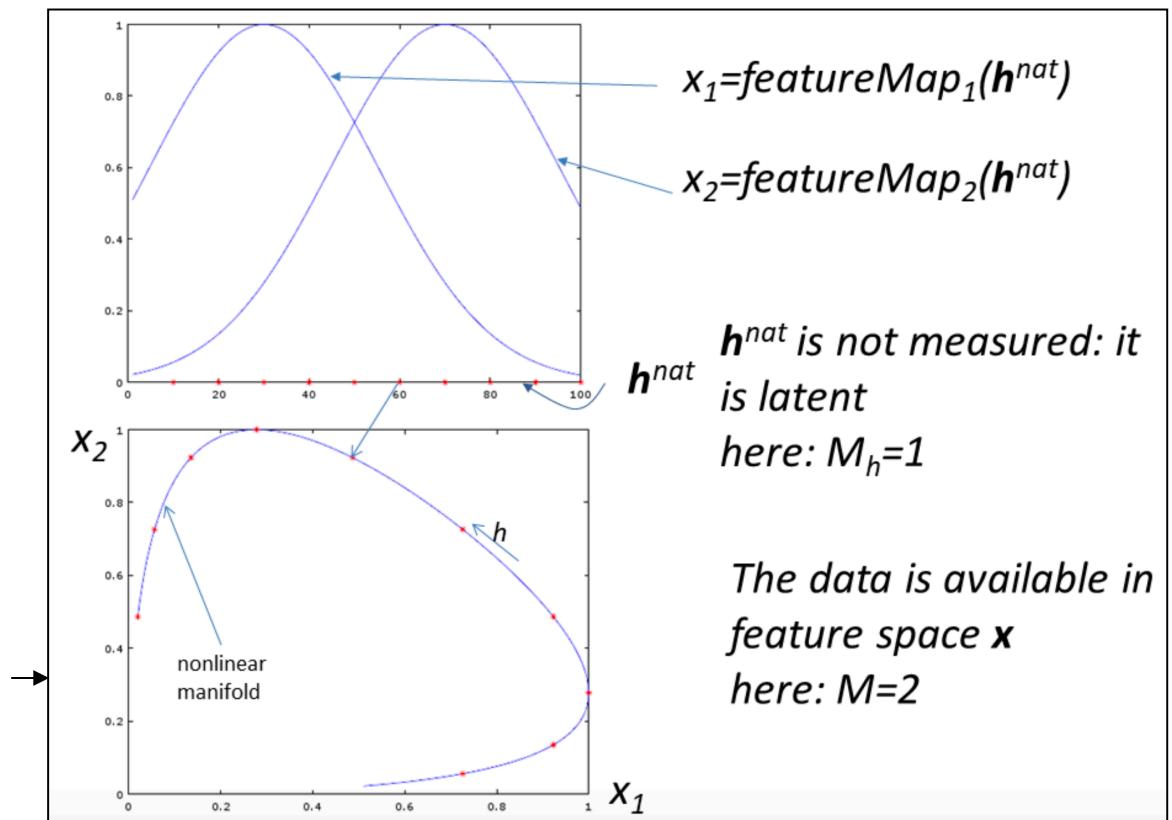
- In the observed M -dimensional input space, the data is distributed on an M_h -dimensional manifold

$$\{\mathbf{x} \in R^M: \quad \mathbf{h} \in R^{M_h} \text{ s.th. } \mathbf{x} = g^{\text{gen}}(\mathbf{h})\}$$

where $g^{\text{gen}}(\cdot)$ is smooth



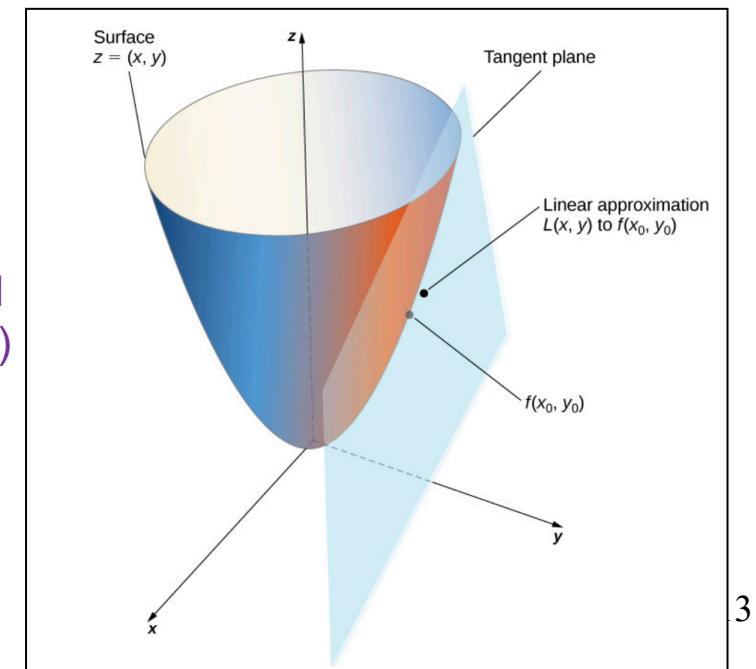
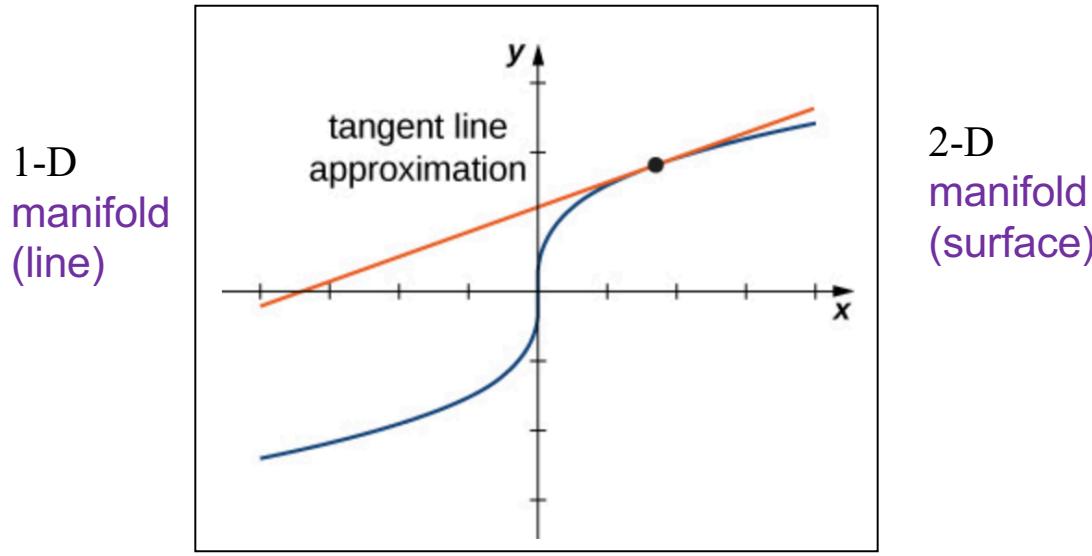
1-D manifold in R^2



Manifolds are specified by Tangent Planes

- Tangents specify how x can change while staying on manifold
 - 1-D: $y = f(x)$ at point $x = x^0$ is given by $y \approx f(x^0) + f'(x^0)(x - x^0)$
 - 2-D: $z = f(x, y)$ at the point (x^0, y^0) is given by

$$z = f(x^0, y^0) + f_x(x^0, y^0)(x - x^0) + f_y(x^0, y^0)(y - y^0)$$
- At a point x on a d -dimensional manifold, the tangent plane is given by d basis vectors that span the local directions of variation allowed on the manifold



Tangents of 1- and 2-D manifolds

- A 1-D manifold in 784-D space (MNIST with 784 pixels)
 - Image is translated vertically
 - Figure below is projection into 2-D space using PCA
 - n -dimensional manifold has n -dimensional plane
 - Tangent is oriented parallel to the surface at that point

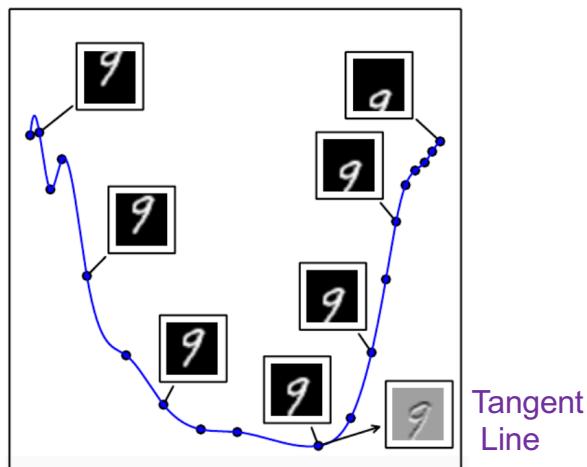
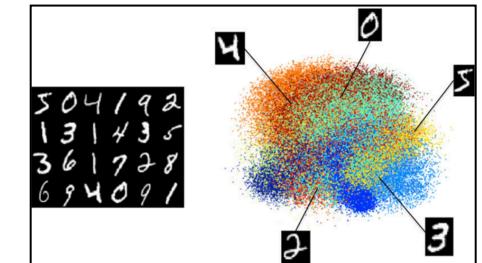
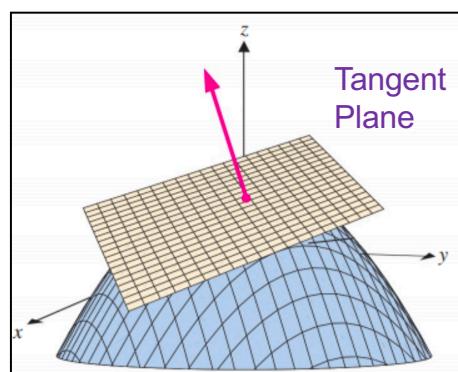


Image shows how this tangent direction appears in image space

Gray pixels indicate pixels that do not change as we move along tangent.

White pixels indicate pixels that brighten, and black those that darken



MNIST
with 3-D PCA

Autoencoder performs trade-off between two forces

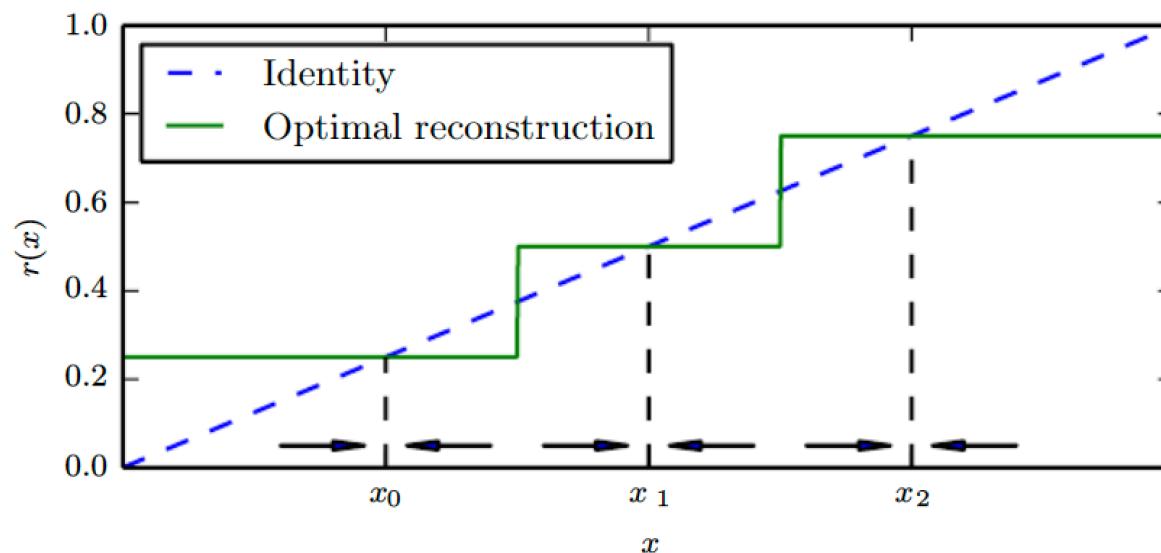
1. Learns representation \mathbf{h} of training example \mathbf{x} such that \mathbf{x} can be recovered through a decoder
 - That \mathbf{x} is drawn from training data is crucial
 - It means the autoencoder need not reconstruct improbable inputs
2. Satisfies the regularization penalty:
 - Limits the capacity of the autoencoder
 - Or it can be a regularization term added to the reconstruction cost
$$L(\mathbf{x}, g(f(\mathbf{x}))) + \Omega(\mathbf{h})$$
 - These techniques prefer solutions less sensitive to input
 - Together they force the hidden representation to capture information about the data generating distribution

What the encoder represents

- Encoder captures only variations needed to reconstruct training examples
- If data generating distribution concentrates near a low-dimensional manifold, this yields representations that implicitly captures a local coordinate system for this manifold
 - Only the variations tangential to this manifold around x need to correspond to changes in $h = f(x)$
 - Hence encoder learns a mapping from input space x to a representation space
 - A mapping that is only sensitive to changes along manifold directions
 - But that is insensitive to changes orthogonal to the manifold

Capturing manifold structure by Invariance

- When reconstruction is insensitive to perturbations around data points, autoencoder recovers manifold structure
 - Ex: 1-D case: manifold is a collection of 0-dimensional manifolds
 - Dashed diagonal line: identity function for target of reconstruction
 - Optimal reconstruction function crosses the identity function whenever there is a data point
 - Horizontal arrows at bottom indicate $r(x)-x$ reconstruction direction vector always pointing towards the nearest “maifold”- a single data point



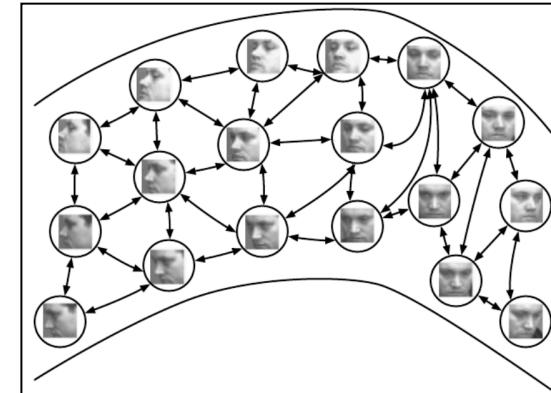
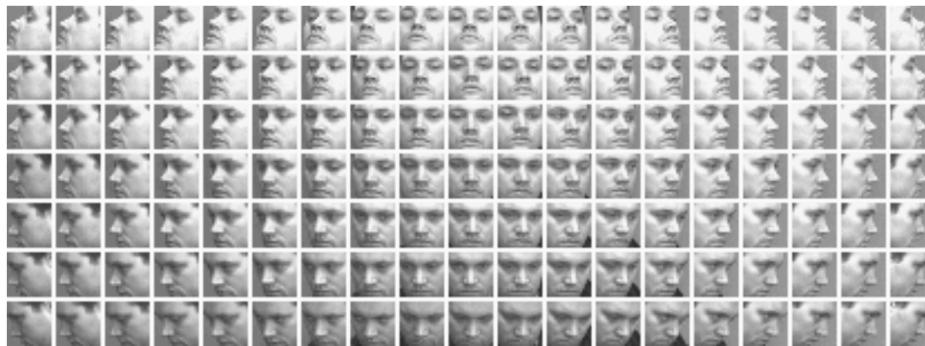
Why autoencoders are useful to learn a manifold

- Compare to other approaches
- Autoencoder
 - Characterizes a manifold
 - Represents data on or near the manifold
 - Representation for a particular example is an *embedding*
 - An embedding has fewer dimensions than the ambient space of which the manifold is a low-dimensional subset
- Other algorithms
 - Non-parametric manifold algorithms
 - Directly learn an embedding for each training example
 - Learn a more general mapping
 - A function to map points in ambient space to embedding

Nonparametric manifold learning

1. Build a nearest-neighbor graph where
 - Nodes represent training examples (one node per sample)
 - Directed edges indicate nearest neighbor relationships
2. Procedures to
 1. Obtain tangent plane associated with a neighborhood of the graph
 2. Associate each training example with an embedding vector
- Works when no of examples is large to cover manifold twists

Queen Mary University of London Multiview Face Dataset



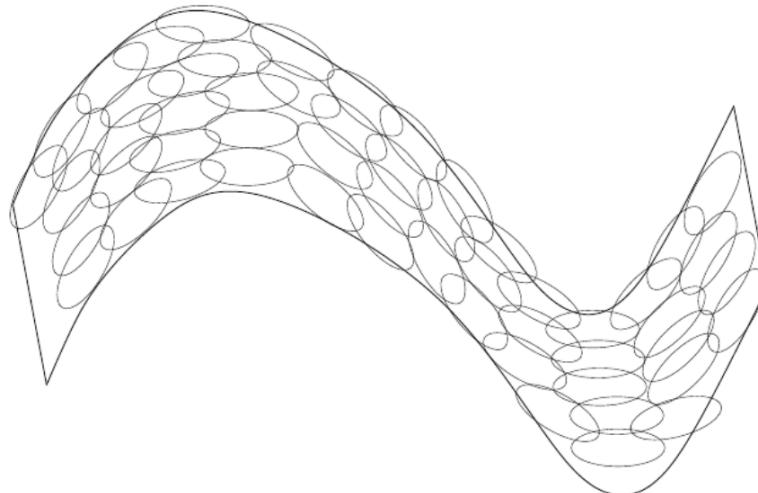
Method associates each node with a tangent plane

One that spans the directions of variations associated with the difference vectors between the example and its neighbors

Tiling a manifold

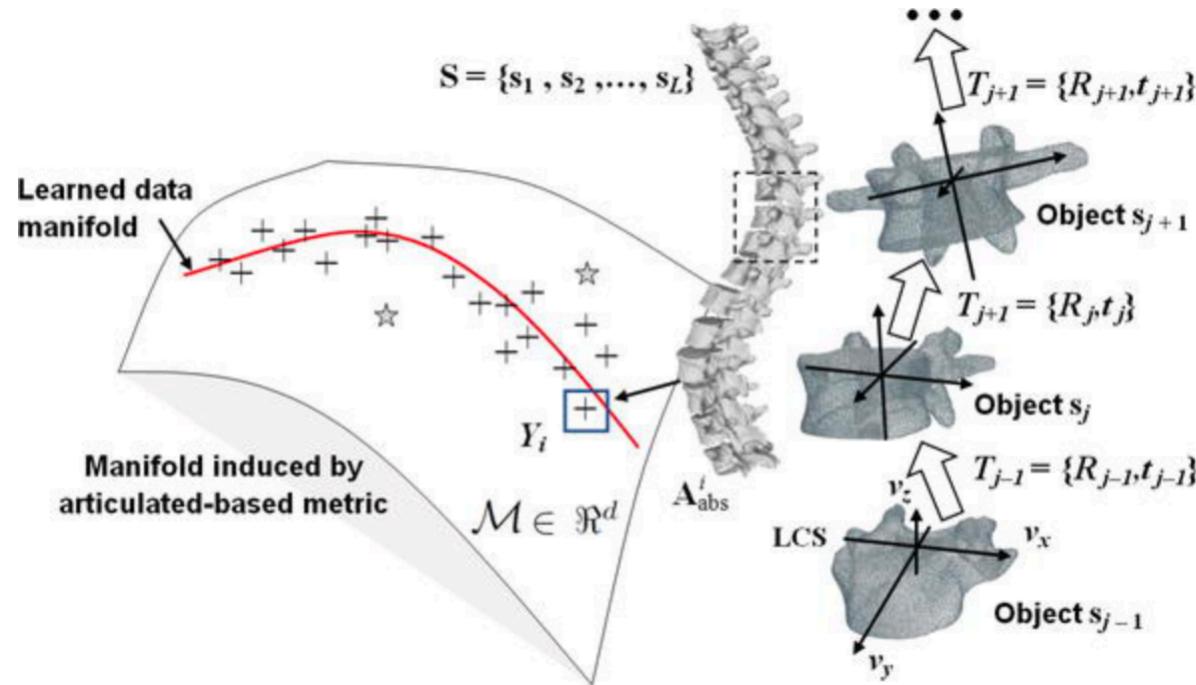
- A global coordinate system can then be obtained through optimization or by solving a linear system
- A manifold can be tiled by a large no of locally linear Gaussian-like patches (or pancakes, because the Gaussians are flat in the tangent directions)

A mixture of Gaussians



- These methods can only generalize the shape of the manifold by interpolating between neighboring examples.
- Unfortunately, manifolds in AI problems are very complicated that can be difficult to capture from only local interpolation

Manifold learning in medical imaging

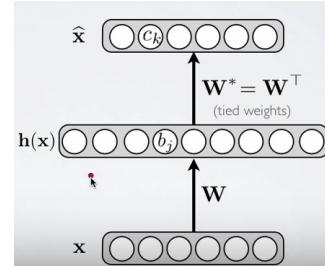


Linear techniques unsuitable for capturing variations in anatomical structures

Structure in the data (CT, MRI, ultrasound) allows a lower dimensional object to describe the degrees of freedom, such as in a manifold structure.

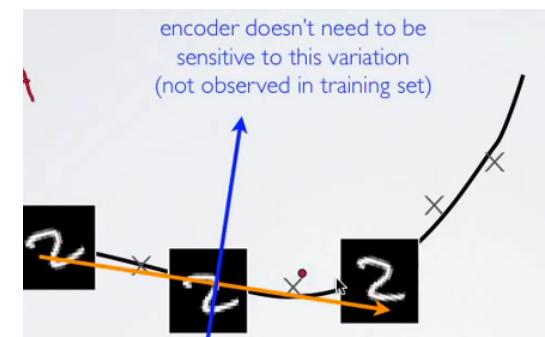
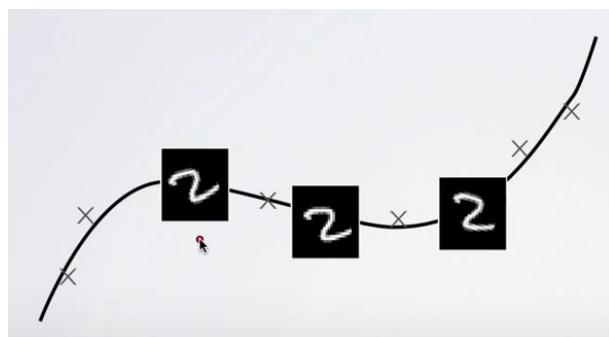
Overcomplete and Contractive Autoencoder

1. Overcomplete



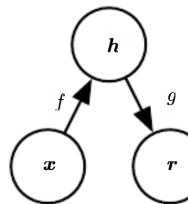
2. Contractive

- Method to avoid uninteresting solutions
- Add an explicit term in the loss that penalizes that solution
- We wish to extract features that only reflect variations observed in the training set
- We would like to be invariant to other variations



Contractive Autoencoder Loss Function

- Contractive autoencoder has an explicit regularizer on $h=f(x)$, encouraging the derivatives of f to be as small as possible:



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

- Where $L(f(x)) + \Omega(h)$
 - Penalty $\Omega(h)$ is the squared Frobenius norm (sum of squared elements) of the Jacobian matrix of partial derivatives associated with encoder function

• New loss function:

$$\underbrace{l(f(\mathbf{x}^{(t)}))}_{\text{autoencoder reconstruction}} + \lambda \underbrace{\|\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})\|_F^2}_{\text{Jacobian of encoder}}$$

► where, for binary observations:

$$l(f(\mathbf{x}^{(t)})) = - \sum_k \left(x_k^{(t)} \log(\hat{x}_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \hat{x}_k^{(t)}) \right)$$

$$\|\nabla_{\mathbf{x}^{(t)}} \mathbf{h}(\mathbf{x}^{(t)})\|_F^2 = \sum_j \sum_k \left(\frac{\partial h(\mathbf{x}^{(t)})_j}{\partial x_k^{(t)}} \right)^2$$

Difference between DAE and CAE

- Denoising Autoencoders make the reconstruction function $r=g(f(x))$ resist small but finite-sized perturbations of the input
 - DAE minimizes $L(x, g(f(\tilde{x})))$
- Contractive Autoencoders make the feature extraction function resist infinitesimal perturbations of the input
 - CAE minimizes $L(f(x)) + \Omega(h)$ where

$$\Omega(h) = \lambda \left\| \frac{\partial f(x)}{\partial x} \right\|_F^2$$
 - It uses a Jacobian-based contractive penalty to pretrain features $f(x)$ for use with a classifier, with $L(x, g(f(x))) + \Omega(h, x)$

$$\Omega(h, x) = \lambda \sum_i \left\| \nabla_x h_i \right\|^2$$

Contractive autoencoder warps space

- The name contractive arises from the way the CAE warps space
- Because CAE is trained to resist perturbations of its input, it is encouraged to map a neighborhood of input points to a smaller neighborhood of output points
 - We can think of this as contracting the input neighborhood to a smaller output neighborhood