COMPSCI 589
Lecture 21: Autoencoders

Benjamin M. Marlin

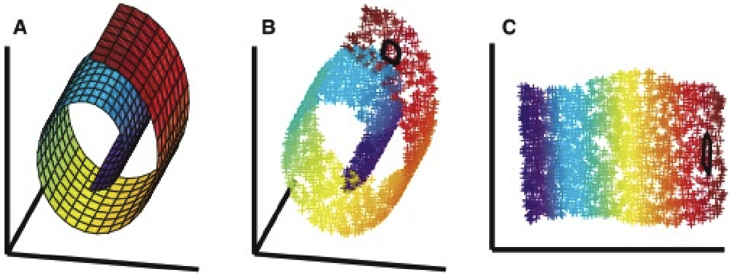
College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).
Created with support from National Science Foundation Award# IIS-1350522.

The Dimensionality Reduction Task

Definition: The Dimensionality Reduction Task

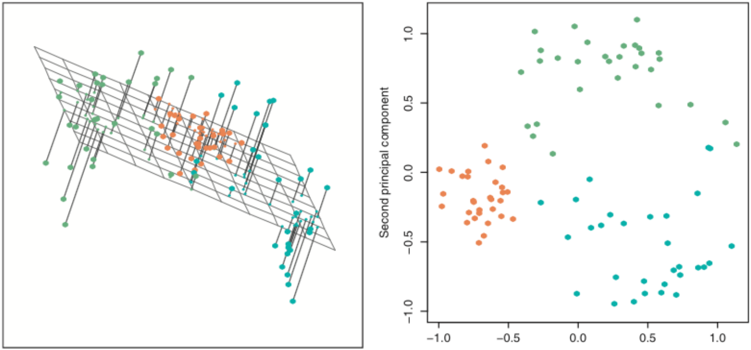
Given a collection of feature vectors $\mathbf{x}_i \in \mathbb{R}^D$, map the feature vectors into a lower dimensional space $\mathbf{z}_i \in \mathbb{R}^K$ where $K < D$ while preserving certain properties of the data.



high-dim distribution | high-dim samples | estimated manifold

Linear Dimensionality Reduction

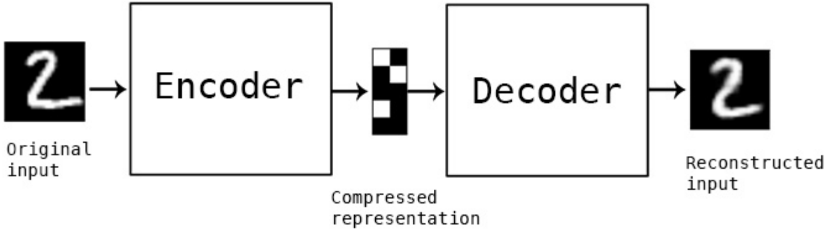
- The simplest dimensionality reduction methods assume that the observed high-dimensional data vectors $\mathbf{x}_i \in \mathbb{R}^D$ lie on a K-dimensional linear manifold within $\mathbb{R}^D$.



Autoencoders

- An autoencoder is a deterministic model that consists of two components: an encoder function and a decoder function.
- The encoder function $f(\mathbf{x})$ maps a D-dimensional input vector $\mathbf{x} \in \mathbb{R}^D$ into a K-dimensional code vector $\mathbf{h} \in \mathbb{R}^K$.
- The decoder function $g(\mathbf{h})$ maps the K-dimensional code vector $\mathbf{h} \in \mathbb{R}^K$ back to a D-dimensional reconstruction of the input vector $\mathbf{r} \in \mathbb{R}^D$.
- A basic goal is to learn to reconstruct $\mathbf{x}$ as the output of the network while constraining $\mathbf{h}$ in a way that forces it to encode salient features of the input while ignoring noise.

Example: Autoencoder for Digits



Linear Autoencoders

- A linear autoencoder is an autoencoder where the encoder and decoder are linear functions.

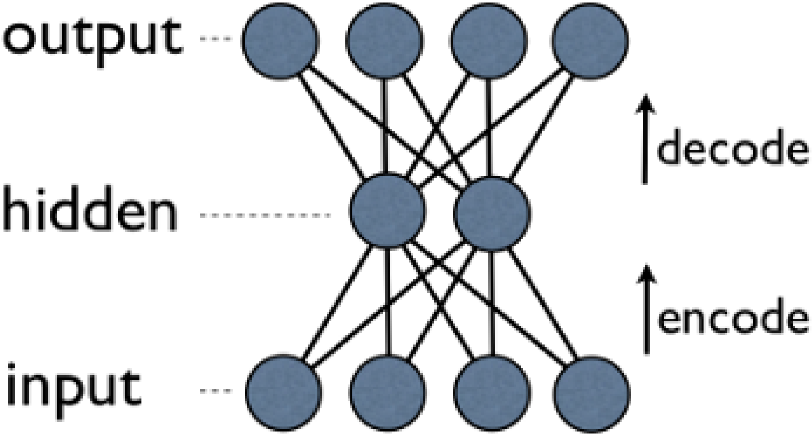
- The encoding and decoding functions are:

$$f(\mathbf{x}) = \mathbf{V}\mathbf{x}$$
$$g(\mathbf{h}) = \mathbf{W}\mathbf{h}$$

- Such a linear encoder-decoder model can be learned by minimizing the MSE between the inputs and the reconstructions, often called the *reconstruction error*.

$$\mathbf{V}^*, \mathbf{W}^* = \arg\min_{\mathbf{V},\mathbf{W}} \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - g(f(\mathbf{x}_n))\|_2^2$$

Example: Basic Linear Autoencoder Model



Constraints on Linear Autoencoders

- In order for an MSE-minimizing linear autoencoder to learn something useful about the structure of the input data, it is necessary to constrain the code vector in some way.
- The most common way to constrain a linear autoencoder is to require $K < D$ (an undercomplete representation).
- This dimensionality reduction forces the model to extract useful information from the inputs and to discard noise in order to minimize the reconstruction error.

Linear Denoising Autoencoders

- Another way to stop an overcomplete autoencoder from simply learning the identity function is to make the problem that the autoencoder has to solve more difficult.

- A *denoising autoencoder* purposely corrupts the input $\mathbf{x}$ using a sample drawn from a stochastic noise process $q(\mathbf{x}'|\mathbf{x})$.

- It then provides $\mathbf{x}'$ as the input to the autoender while requiring the reconstruction that the model produced match the original $\mathbf{x}$.

Linear Denoising Autoencoders

- The model can be trained as shown below. The characteristics of the noise process are hyperparameters of model.

$$\mathbf{V}^*, \mathbf{W}^* = \underset{\mathbf{V},\mathbf{W}}{\arg\min} \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{x}'|\mathbf{x}_n)} \left[\|\mathbf{x}_n - g(f(\mathbf{x}'))\|_2^2 \right]$$

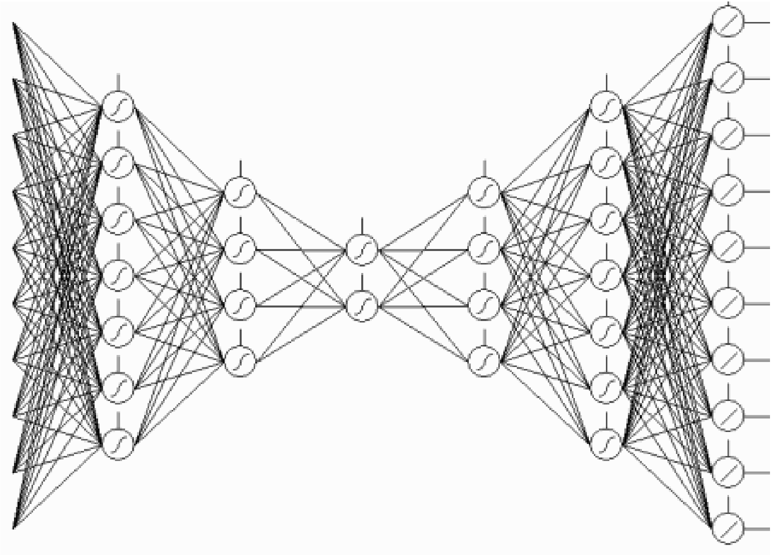
- To implement this approach, on every learning iteration we sample $\mathbf{x}'_n$ from the noise distribution $q(\mathbf{x}'|\mathbf{x}_n)$ and use automatic differentiation to compute the gradient shown below:

$$\frac{1}{N} \sum_{n=1}^{N} \nabla \|\mathbf{x}_n - g(f(\mathbf{x}'_n))\|_2^2$$

Non-Linear Autoencoders

- All of the above models and training criteria can only accurately represent data defined on linear manifolds.
- To make an autoencoder non-linear, it suffices to make the encoder and decoder networks non-linear.
- However, the capacity of non-linear autoencoders is no longer limited by the length of the code vector.

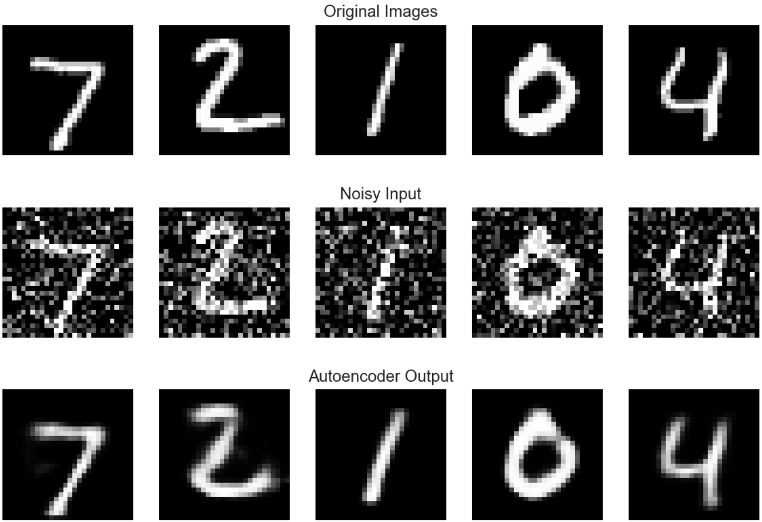
Example: Deep Autoencoders



Constraining Non-Linear Autoencoders

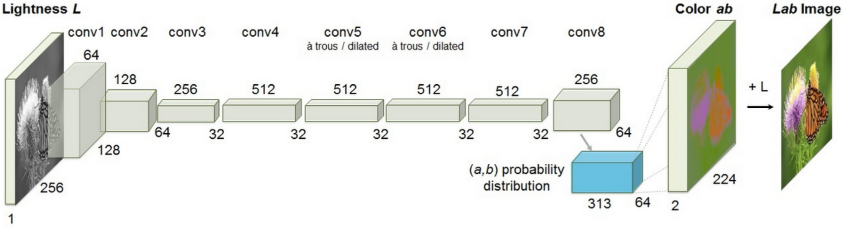
- Nonlinear autoencoders need to have constraints to force them to extract useful information from the data.
- Possible constraints include constraining the width and depth of the encoder and decoder networks and using various forms of the denoising principle.
- As with supervised learning, it is known that some data distributions can be much more efficiently represented with deep, non-linear autoencoders instead of shallow non-linear or linear autoencoders.
- The key is determining the network architecture that gives the best performance for a given down-stream task.

Deep Denoising Autoencoder for Images

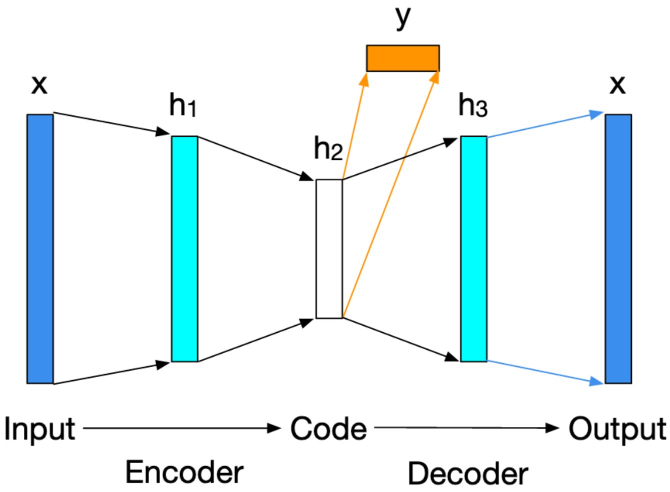


Deep Autoencoder for Image Colorization

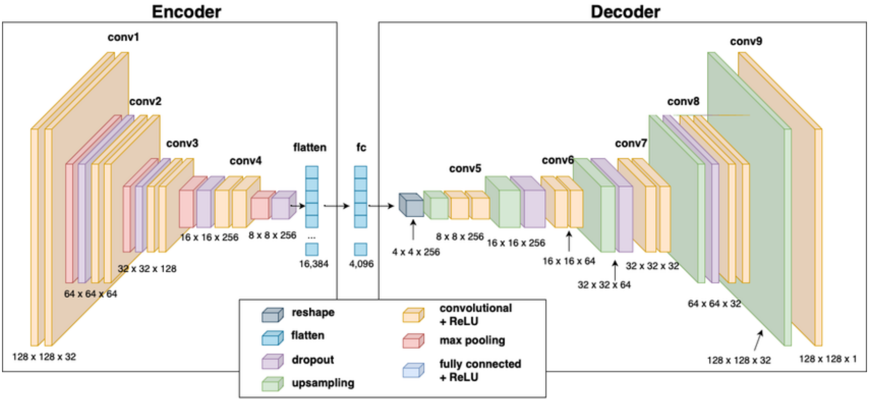
Figure 13: The CAE is trained to colorize the image



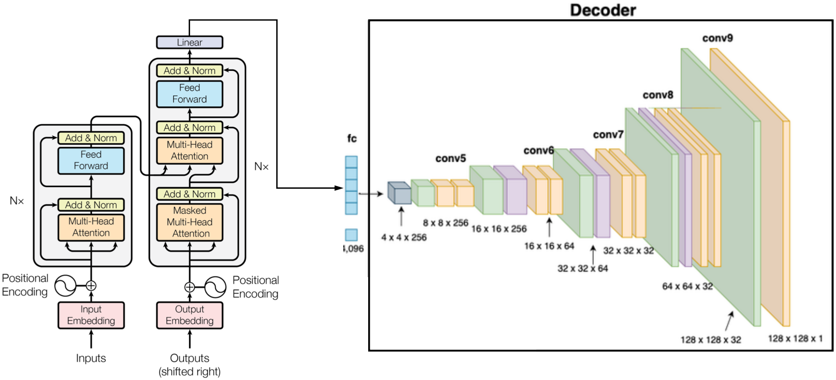
Semi-Supervised Autoencoder/Classifier



Text Conditioned Image Generation



Text Conditioned Image Generation



Text Conditioned Image Generation

"a green train is coming down the tracks" "a group of skiers are preparing to ski down a mountain." "a small kitchen with a low ceiling" "a group of elephants walking in muddy water." "a living area with a television and a table"

https://arxiv.org/pdf/2204.06125.pdf

(Dall-e 2 Paper, April 2022)

Text Conditioned Image Generation

Teddy bears swimming at the Olympics 400m Butterfly event.

A cute corgi lives in a house made out of sushi.

A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

A brain riding a rocketship heading towards the moon.

A dragon fruit wearing karate belt in the snow.

A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.

`https://arxiv.org/pdf/2205.11487.pdf`

(Imagen Paper, May 2022)

Text Conditioned Image Generation

A fierce garden gnome warrior, clad in armor crafted from leaves and bark, brandishes a tiny sword and shield. He stands valiantly on a rock amidst a blooming garden, surrounded by colorful flowers and towering plants. A determined expression is painted on his face, ready to defend his garden kingdom.

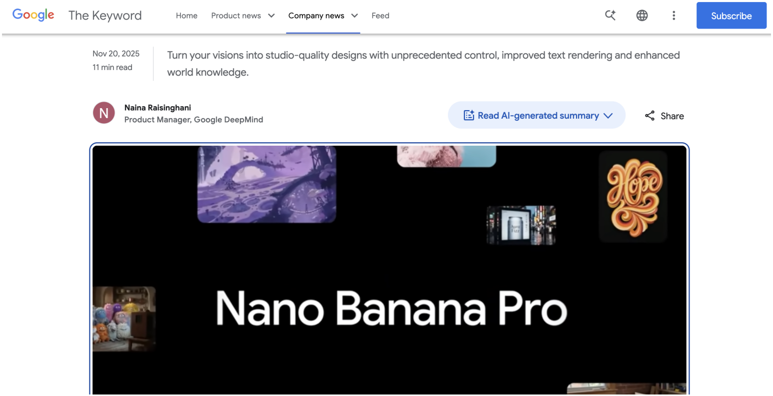
An icy landscape under a starlit sky, where a magnificent frozen waterfall flows over a cliff. In the center of the scene, a fire burns bright, its flames seemingly frozen in place, casting a shimmering glow on the surrounding ice and snow.

A swirling, multicolored portal emerges from the depths of an ocean of coffee, with waves of the rich liquid gently rippling outward. The portal engulfs a coffee cup, which serves as a gateway to a fantastical dimension. The surrounding digital art landscape reflects the colors of the portal, creating an alluring scene of endless possibilities.

`https://cdn.openai.com/papers/dall-e-3.pdf`

(Dall-e 3 Paper, Oct 19, 2023)

Text Conditioned Image Generation



https://blog.google/technology/ai/nano-banana-pro/
https://deepmind.google/models/imagen/
(Nano Banan Pro / Imagen 4, Nov 20, 2025)

Text Conditioned Video Generation

Wooden figurine surfing on a surfboard in space.

Balloon full of water exploding in extreme slow motion.

Melting pistachio ice cream dripping down the cone.

https://arxiv.org/pdf/2210.02303.pdf

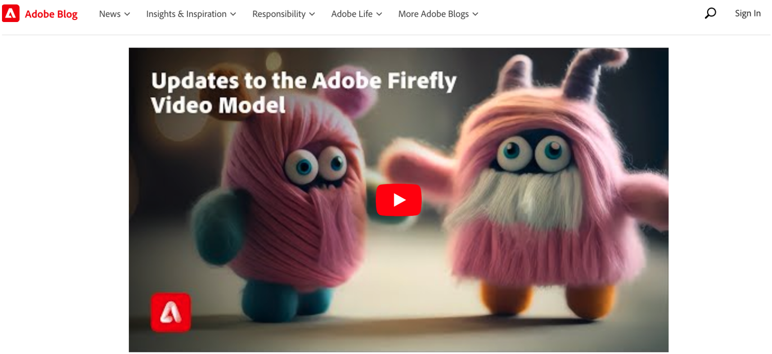
https://imagen.research.google/video

(Imagen Video Paper, Oct 2022)

Text Conditioned Video Generation

https://stability.ai/stable-video
(Stable Diffusion Video, Nov 21, 2023)

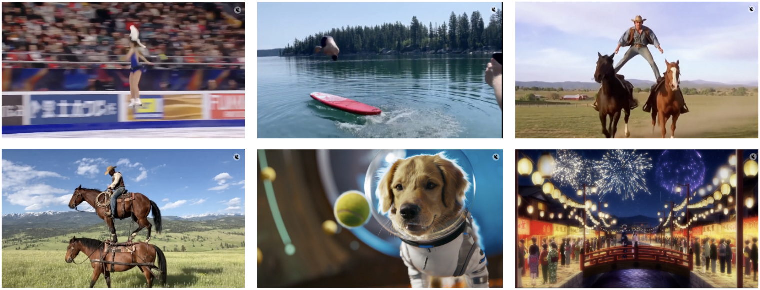
Text Conditioned Video Generation



https://blog.adobe.com/en/publish/2024/09/11/bringing-gen-ai-to-video-adobe-firefly-video-model-coming-soo

(Adobe Firefly, Sept 11, 2024)

Text Conditioned Video Generation



`https://openai.com/index/sora-2/`

(Open AI Sora 2, Sept 30, 2025)