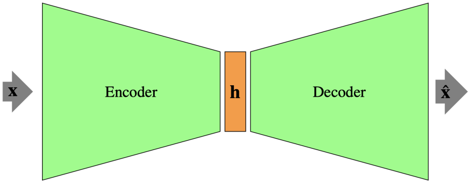
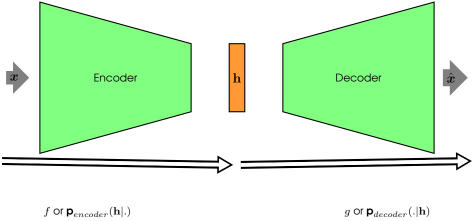


Autoencoders

Vincent Barra
LIMOS, UMR 6158 CNRS, Université Clermont Auvergne

What is an autoencoder ?



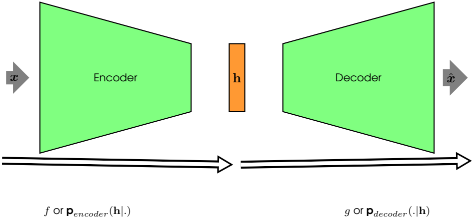
Key ideas

- A neural network trained using unsupervised learning
- Trained to copy its input to its output
- Learns an embedding h

$$\hat{\boldsymbol{x}} = g[f(\boldsymbol{x})] \quad h = f(\boldsymbol{x})$$

LIMOS Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes

What is an autoencoder ?



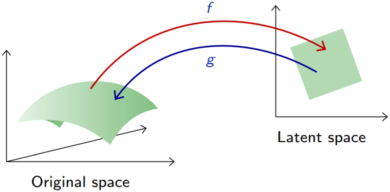
Key ideas

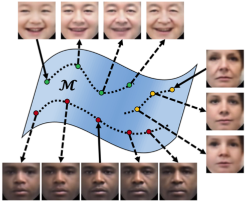
- $f : \mathbb{R}^n \rightarrow \mathbb{R}^q$: Encoder (MLP, CNN, RNN,..)
- $g : \mathbb{R}^q \rightarrow \mathbb{R}^n$: Decoder (MLP, CNN, RNN,..)
- $h = f(\boldsymbol{x})$: Encoded representation of x into a latent space of dimension q
- $q > n$: overcomplete / $q < n$: undercomplete

Latent space

Undercomplete autoencoder

$(\forall \boldsymbol{x})\ f(\boldsymbol{x})$ lies onto a manifold in $\mathbb{R}^q$



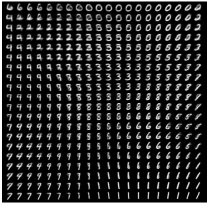


LIMOS — Laboratoire d'informatique, de modélisation et d'optimisation des systèmes
CNRS
UCA Université Clermont Auvergne

What is learned ?

Learn relevant features

- Learning $\hat{\boldsymbol{x}} = \boldsymbol{x}$ is not useful ...
- Autoencoders are designed to be unable to copy perfectly
- Forced to capture most salient features of training data
- So... what to retain ?
- f and g can be probabilistic mappings $\mathbf{p}_{encoder}(\mathbf{h}|.)$ and $\mathbf{p}_{decoder}(.|\mathbf{h})$

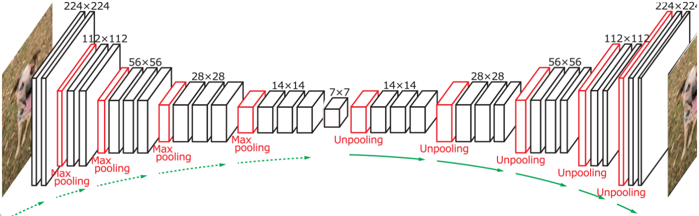


LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

What is learned ?

What/how to learn ?

- Autoencoders are composed of two neural nets
- $\Rightarrow f = f_{\boldsymbol{W}_1,\boldsymbol{b}_1}$ and $g = g_{\boldsymbol{W}_2,\boldsymbol{b}_2}$
  - As usual, definition of a loss function to be minimized
  - Gradients computation using backpropagation
- Can also be trained using recirculation
  - Compare activations on $\boldsymbol{x}$ to activations of $\hat{\boldsymbol{x}}$
  - Biologically plausible than backpropagation, rarely used



LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

UNIVERSITÉ Clermont Auvergne

Loss function

Loss function

$$\min_{W_1, \boldsymbol{b}_1, W_2, \boldsymbol{b}_2} L(\boldsymbol{x}, g[f(\boldsymbol{x})]) + \lambda \Omega(\boldsymbol{h})$$

- L: penalizing $g[f(\boldsymbol{x})])$ for being dissimilar from $\boldsymbol{x}$
- $\Omega(\boldsymbol{h})$: Regularization term

Regularization

Allows

- Sparsity of representation
- Robustness to missing inputs
- Robustness to noise
- ...

A simple example

Encoder = Decoder = 1 hidden layer MLP

$$\boldsymbol{x} \in \mathbb{R}^n \underset{f}{\longrightarrow} \boldsymbol{h} \in \mathbb{R}^q \underset{g}{\longrightarrow} \hat{\boldsymbol{x}} \in \mathbb{R}^n$$

σ_1, σ_2: element-wise real activation functions, $W \in \mathcal{M}_{qn}(\mathbb{R})$, $b_1 \in \mathbb{R}^q$, $b_2 \in \mathbb{R}^n$

$$f(\boldsymbol{x}) = \sigma_1(W\boldsymbol{x} + b_1) \quad \hat{\boldsymbol{x}} = \sigma_2(W^T\boldsymbol{h} + b_2)$$

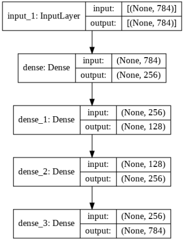
An autoencoder without regularization is trained to minimize

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 = \|\boldsymbol{x} - \sigma_2(W^T(\sigma_1(W\boldsymbol{x} + b_1) + b_2)\|^2$$

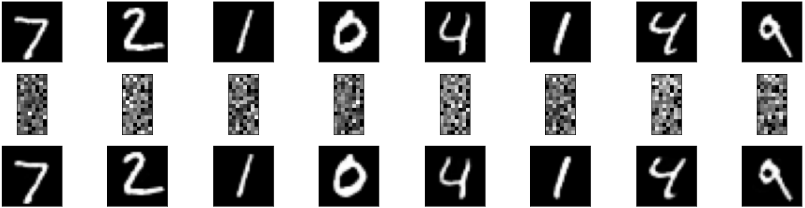
If σ_2 is linear and $q < n$, this autoencoder is equivalent to PCA.

Adding a layer...

Encoder = Decoder = 2 hidden layers MLP.. what you will have to do...



$\sigma_{Encoder}$ = {Relu,Relu}
$\sigma_{Decoder}$ = {Relu,sigmoid}
L = MSE
$Optimizer$ = Adam.



Encoder/decoder capacity

Not too much capacity

- too much capacity for f and $g \Rightarrow$ autoencoder can learn identity without learning any useful information about distribution of data
- if the encoder is very powerful, then $q = 1$ is possible and the decoder can learn to map back to the values of specific training examples

Failure if...

- capacity too high (controled by depth)
- $q \geq n$ (is no strong constraint is applied on the loss)

$\Rightarrow$ Need for regularization.

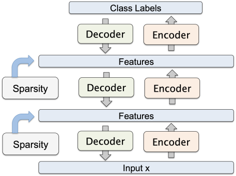
LIMOS — Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes

Université Clermont Auvergne

Representation power

Deep Encoder/Decoder can provide many advantages.
Common strategy: greedily pretrain a stack of shallow autoencoders

- The first layer is trained on input data $\rightarrow W_1, b_1$
- Input $\rightarrow$ vector of the activation values of the hidden layer
- The second layer is trained on this vector $\rightarrow W_2, b_2$
- Repeat the process



Regularization

Regularized autoencoders can help

- Rather than limiting model capacity by keeping encoder/decoder shallow and q small
- Use of $\Omega(\boldsymbol{h})$ in the loss function
- Forces the autoencoder to have more properties than the simple identity

Some properties

- Sparsity of representation
- Robustness to missing inputs
- Robustness to noise
- Even in $q \geq n$ can learn useful information
- ...

Regularization

Sparse autoencoder

- Only a few neurons are proned to activate when a single sample is input into the network
- Can be done using $\Omega(\boldsymbol{h}) = \|W\|_1$ (L_1 norm)
- Can be done using the Kullback-Leibler divergence

Contractive autoencoder

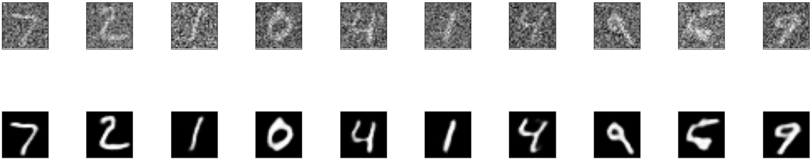
- $\Omega(\boldsymbol{h}) = \sum_{i=1}^{q} \|\nabla_x h_i\|^2$
- Learns a function that doesn't change much when $\boldsymbol{x}$ slightly changes
- Warps space: resists to perturbations of its input
- Encourages to map a neighborhood of input $\boldsymbol{x}$ to a smaller neighborhood of output points

Denoising autoencoders

Receive a corrupted data as input and trained to predict the original, uncorrupted data as its output

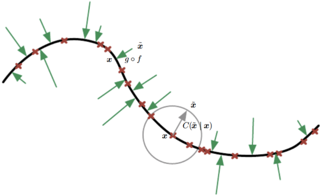
$$Min\ L(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = -log\ \mathsf{p}_{encoder}(\boldsymbol{x}|\boldsymbol{h} = f(\boldsymbol{x})) \quad \tilde{\boldsymbol{x}} \text{ corrupted version of } \boldsymbol{x}$$

Performs SGD on $\mathbb{E}_{\tilde{\boldsymbol{x}}} log\ \mathsf{p}_{decoder}(\boldsymbol{x}|\boldsymbol{h} = f(\tilde{\boldsymbol{x}}))$



Denoising autoencoders

- Can learn a corruption process $C(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ and the corresponding denoising process.
- Learns a vector field: the training samples lie on a low-dimensional manifold. The vector field estimates the slope of the density of data



- red crosses: training examples
- grey circle: equiprobable corruption of a training example
- green arrows: vector field

Denoising autoencoders

Vector field

Encouraging the model to have the same score $S = \nabla_x log\ \mathbf{p}(\boldsymbol{x})$ as the data distribution at every training point $\boldsymbol{x}$

Denoising autoencoder with Gaussian $\mathbf{p}(\boldsymbol{x} \mid \boldsymbol{h})$ estimates the score as

$$S \approx g[f(\boldsymbol{x})] - \boldsymbol{x}$$

and is trained to minimize

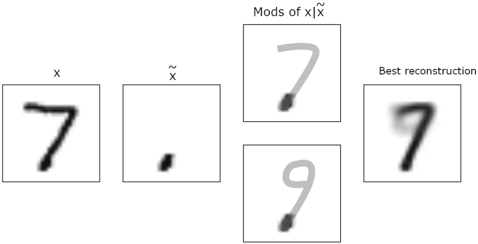
$$\|g[f(\tilde{\boldsymbol{x}})] - \boldsymbol{x}\|^2$$

LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

Denoising autoencoders

The posterior $\mathbf{P}_{\boldsymbol{x}|\tilde{\boldsymbol{x}}}$ can be non-deterministic.

If $L(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2$: best reconstruction is $\mathbb{E}(\boldsymbol{x} \mid \tilde{\boldsymbol{x}})$: very unlikely under $\mathbf{P}_{\boldsymbol{x}|\tilde{\boldsymbol{x}}}$



Adversarial networks

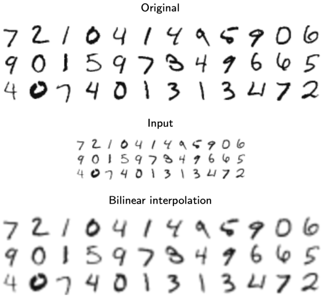
Use in place of loss a second network that assesses if the output is realistic

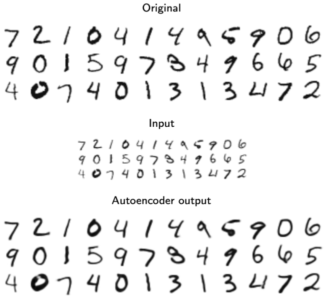
Additional ressource

See lecture “Generative models (GAN)”.

Super resolution

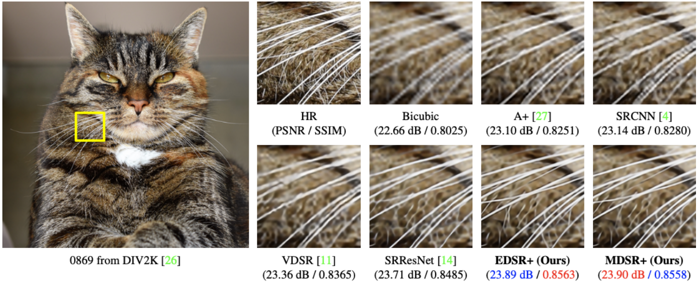
Special case of denoising autoencoders, where the encoder's input is smaller thant the decoder output.





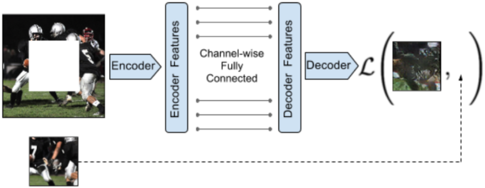
Super resolution

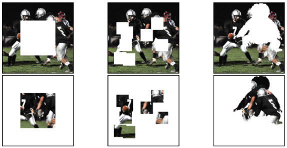
For super resolution on real images, use deep nets (here 2 different ResNets, Lim et al (2017))



Context Encoders

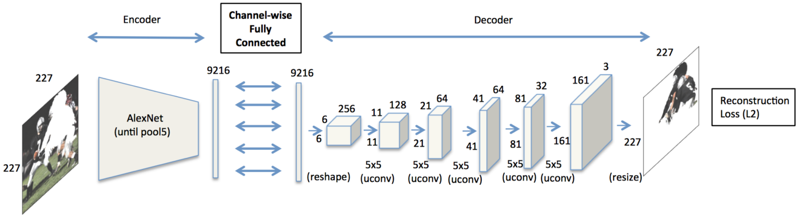
An interesting work of Pathak et al., 2016



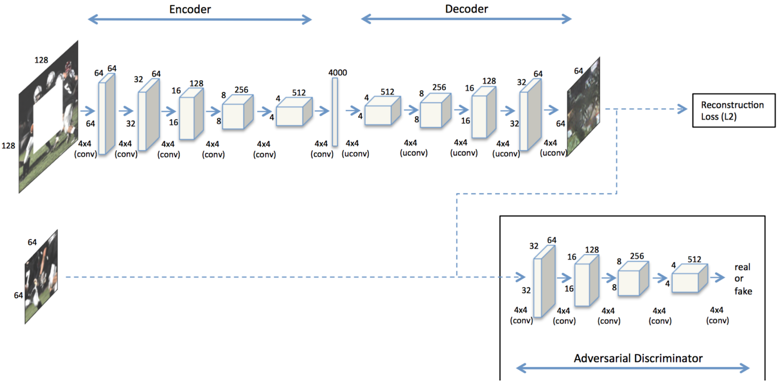


(a) Central region (b) Random block (c) Random region

Context Encoders



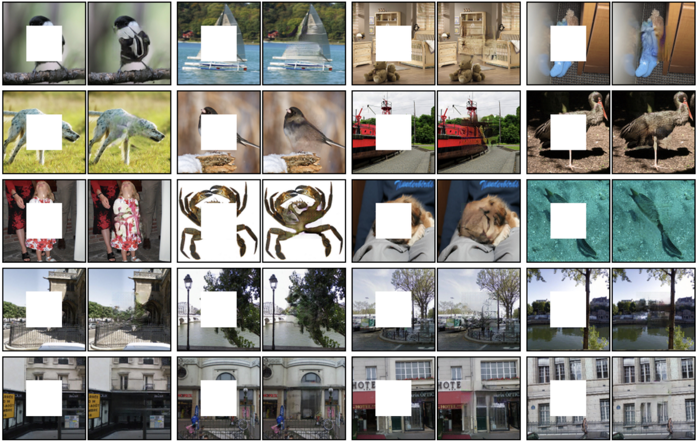
Context Encoders



Additional ressource

See lecture “Generative models (GAN)”.

Context Encoders



Latent space

Can be used in several ways:

- Once learned, the Encoder provides a concise way to compress the data, while retaining relevant information

$\Rightarrow$ input to classic machine learning methods

- Directly working in the latent space : analysis, generation with the decoder...

Additional ressource

See lecture "Generative models (VAE)".

Manifold in Latent space

Hypothesis

$f(\forall \boldsymbol{x})\ (\boldsymbol{x})$ concentrates around a low dimensional manifold $\mathcal{M}$ in $\mathbb{R}^q$.

"Justification"

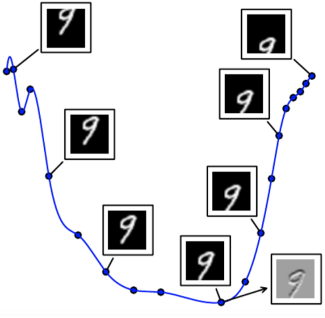
- $\boldsymbol{x}$ = image of size $m \times n$, each pixel being coded between 0 and 255.
- Set of all imges : $256^{m \times n}$ possible images.
- "Cat" images" impose constraints on grey level distribution

$\Rightarrow$ Less degrees of freedom in the high dimensional space

Autoencoders aim to learn the structure of $\mathcal{M}$.

Manifold in Latent space

Manifolds are defined by tangent planes: for $x \in \mathcal{M}$, specifies how x can change while staying on $\mathcal{M}$.



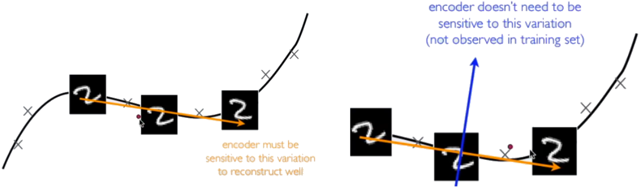
Moving along tangent

- gray pixels: don't change
- white pixels brighten
- black pixels darken

LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES
cnrs
UCA UNIVERSITÉ Clermont Auvergne

Manifold in Latent space

- Encoder captures the information needed to reconstruct $\boldsymbol{x}$ $\rightarrow$ representation $\boldsymbol{h}$
- If data generating distribution concentrates near a low- dimensional $\mathcal{M}$, $\boldsymbol{h}$ implicitly captures a local coordinate system for $\mathcal{M}$
  - Only variations tangential to $\mathcal{M}$ at $\boldsymbol{x}$ need to correspond to changes in $\boldsymbol{h}$
  - The learned f is only sensitive to changes in the tangent plane, not to orthogonal changes.



LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES
cnrs
UCA UNIVERSITÉ Clermont Auvergne

Manifold in Latent space

An example of application: interpolation between images