Autoencoding

Dieng Bara, Leber Stevan, Pereira Nolwenn, Florian Robino

December 2019

M2 - AIDN Deep Learning

Summary

1. Autoencoders - Basics
2. Sparse coding
3. Stacked autoencoders
4. Restricted Bolzmann Machine
5. Denoising autoencoders
6. Contractive autoencoders
7. Variational autoencoders

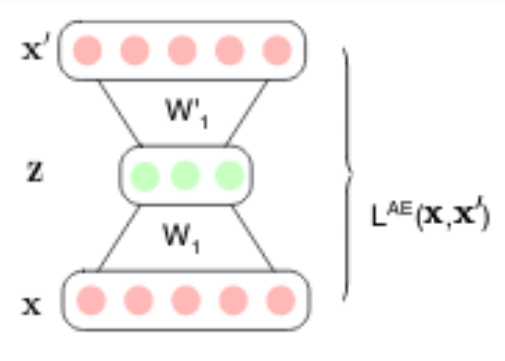
Autoencoders - Basics

Autoencoders (AE)

Single-layer feed-forward neural network :

- input layer
- hidden layer
- output layer

Input and output layers have the same size



Green layer = bottleneck, lossy compressed representation

Autoencoders (AE)

Goal : reconstruct its original inputs

Non-linear transformation

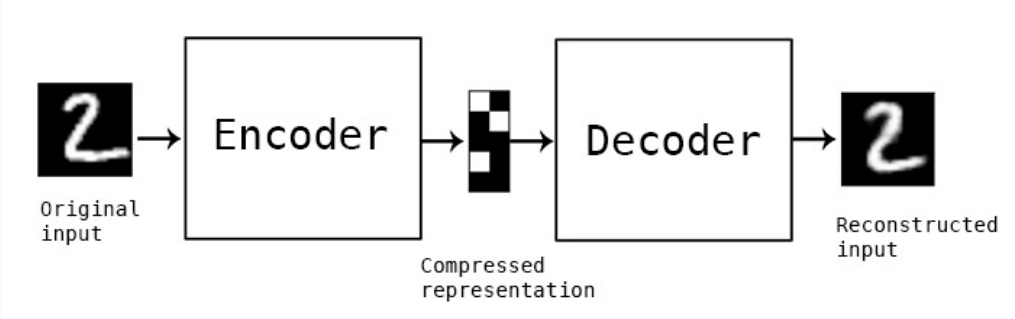
Examples of encoders with linear transformation :

- JPEG (DCT)
- PCA

Autoencoders (AE)

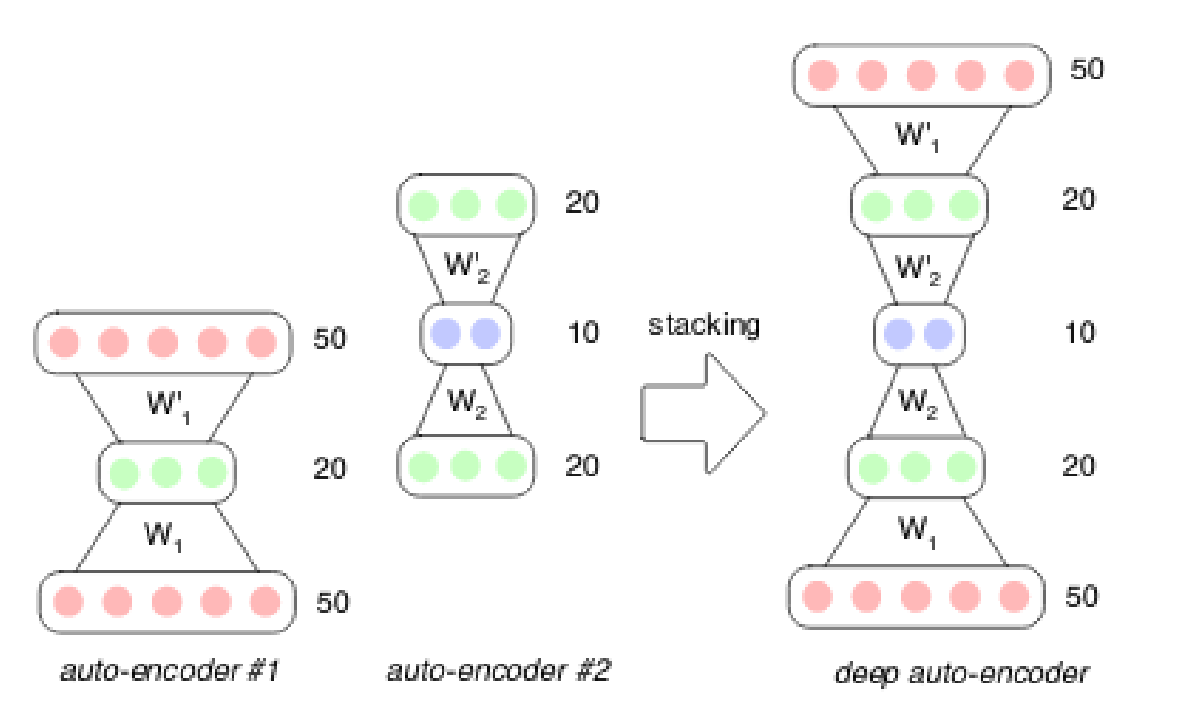
Most of the time, type of activation function in the encoder and decoder of an autoencoder should be the same.

$$L(x, x') = ||x - x'||^2 \tag{1}$$



Autoencoders (AE)

Deep autoencoders : autoencoders with hidden layers



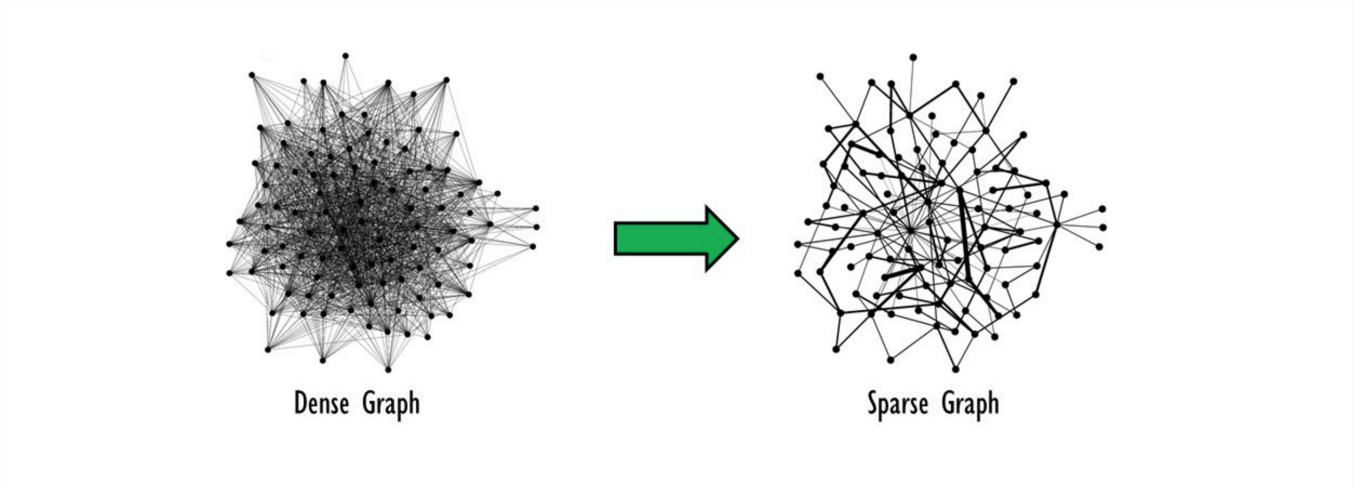
Sparse coding

Sparse coding

Dense : a feature activates half of the neurons

Local : a feature activates a single neuron

Sparse : between local and dense. We want to activate the useful neurons



Sparse coding

Comparable to :

- Decomposing all the matters in the universe in a composition of atoms
- Decomposing music in a composition of just a few notes

We avoid interferences (when one neuron reacts to two different features, we dont know what happens if they're active at the same time)

Dimensions of hidden layer will be larger than input layer

Sparse coding

Producing sparse binary spaces :

- regularization terms making the codes sparse and well distributed
- binary relaxation term forcing the activation of the middle layer to values as close as possible to the binary value b

Sparse coding

b bits should be distributed uniformly in the hashing layer but in practice the network tends to focus on the same bits to encode all images. To produce more balanced bits, we maximize the variance of each bit and force the bits to be pairwise uncorrelated

$$\underset{W,d}{minimize}\ l_2 = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{d} |||b_i^j| - 1|| \tag{2}$$

with b_j activation vector of the hashing layer for an image j and d the dimension of the code

Sparse coding

Fault tolerance, correction of error is easier than local code since more than one neuron is activated per stimulus

Sparse penalty (Ω) applied to the encoding parameters θ_e :

$$J(\theta, x) = L(x, g_{\theta_d}(f_{\theta_e}(x)) + \lambda \times \Omega(\theta_e) \tag{3}$$

Respond to unique statistical features of the dataset it has been trained on

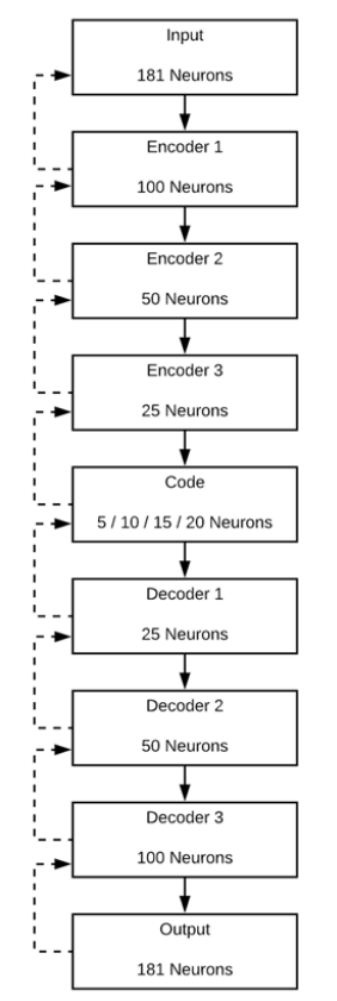
Stacked autoencoders

Stacked autoencoders

Stacking multiple standard autoencoders :

- The first autoencoder is trained using raw data as input.
- The next autoencoders are trained using the learned representation produced by the former autoencoder as its input data.
- SAE is constructed by combining the input layer and all hidden layers of those trained standard autoencoders.

Stacked autoencoders

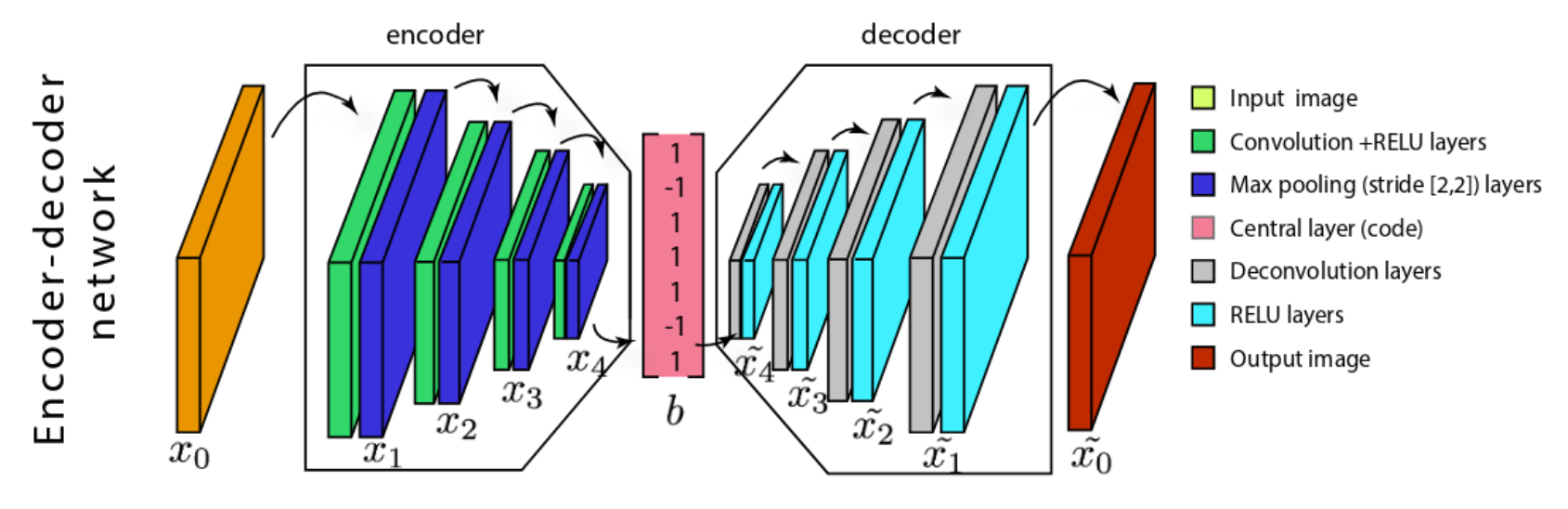


Stacked autoencoders

Number of Neuron	Epochs						
	50	100	300	500	1000	1500	1700
5	74.60 %	78.57 %	73.81 %	78.57 %	82.54 %	80.16 %	84.92 %
10	74.60 %	82.54 %	77.78 %	76.19 %	82.54 %	84.92 %	78.57 %
15	76.98 %	82.54 %	73.81 %	82.54 %	83.33 %	80.95 %	80.16 %
20	76.19 %	81.75 %	76.19 %	76.98 %	82.54 %	**86.51 %**	84.13 %

Usually : relu for hidden layers and sigmoid for output

Stacked autoencoders



Stacked convolutional autoencoder :

- encoder : conv, relu, batch_normalization, pooling
- decoder : unpooling, conv, relu

Restricted Bolzmann Machine

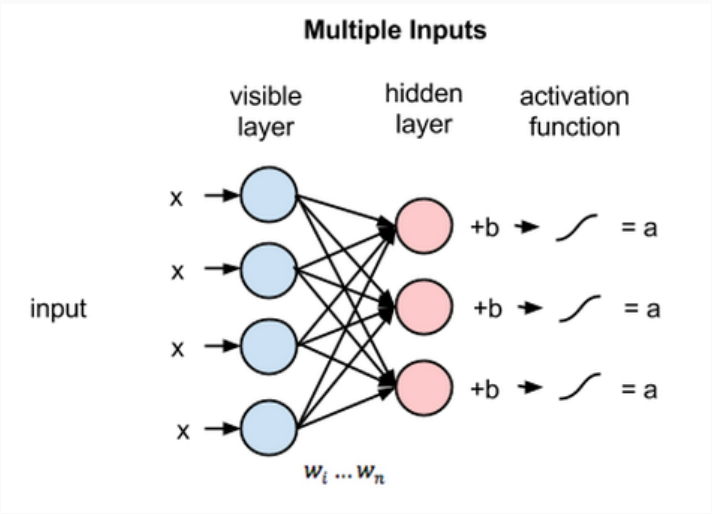
Restricted Bolzmann Machine

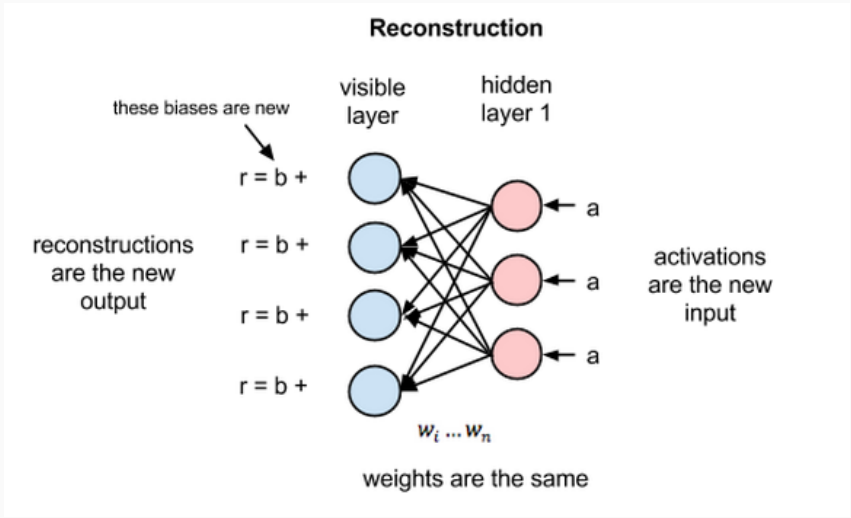
Created by Geoff Hinton

Between an input and a hidden layer : choose to transmit the input or not. Used for pretraining

Goal : recreate the inputs as accurately as possible

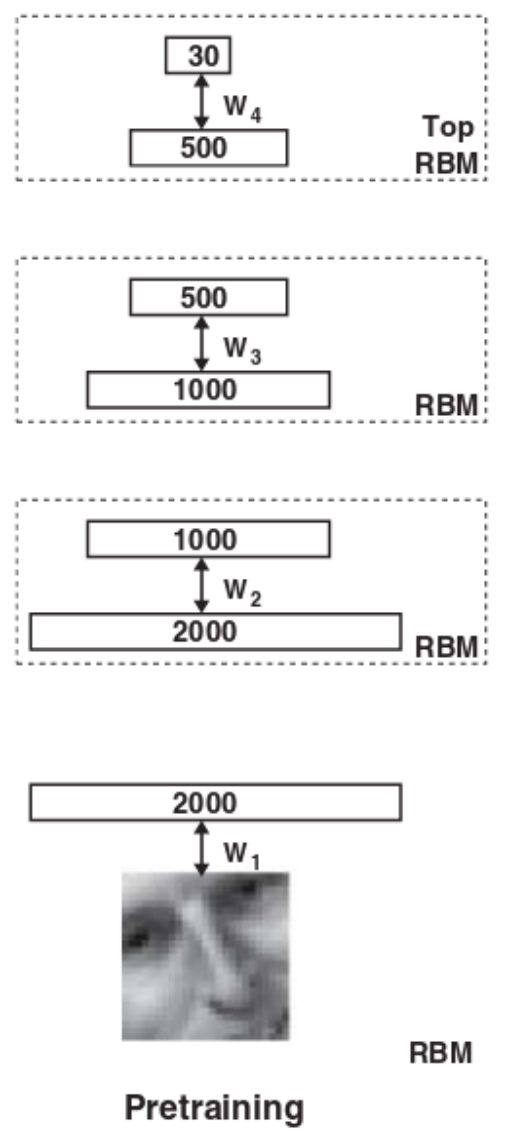
Restricted Bolzmann Machine





- forward pass : the inputs are modified by weights and biases and are used to activate the hidden layer
- next pass,the activations from the hidden layer are modified by weights and biases and sent back to the input layer for activation
- comparaison with the original input

Restricted Bolzmann Machine



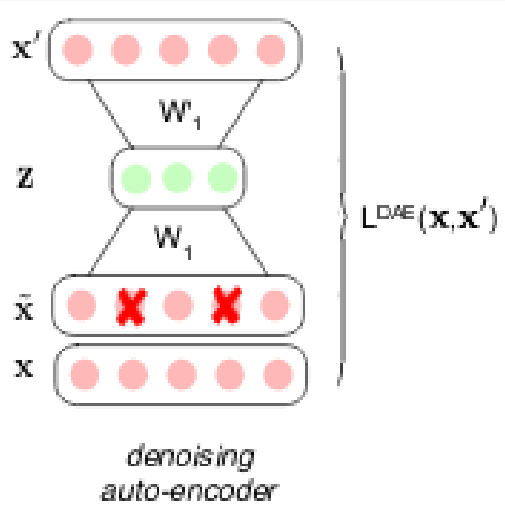
Denoising autoencoders

Denoising autoencoders (DAE)

Variant of the autoencoder :

- x is replaced by $\tilde{x}$
- $\tilde{x}$ corrupted version of x

loss function is not written as a function of the corrupted input vector $\tilde{x}$, but to its uncorrupted version x.



Denoising autoencoders (DAE)

Why denoising ? We want to have a stable network that resists to corruption and that extracts the useful structures. It fixes a corrupted input $\tilde{x}$.

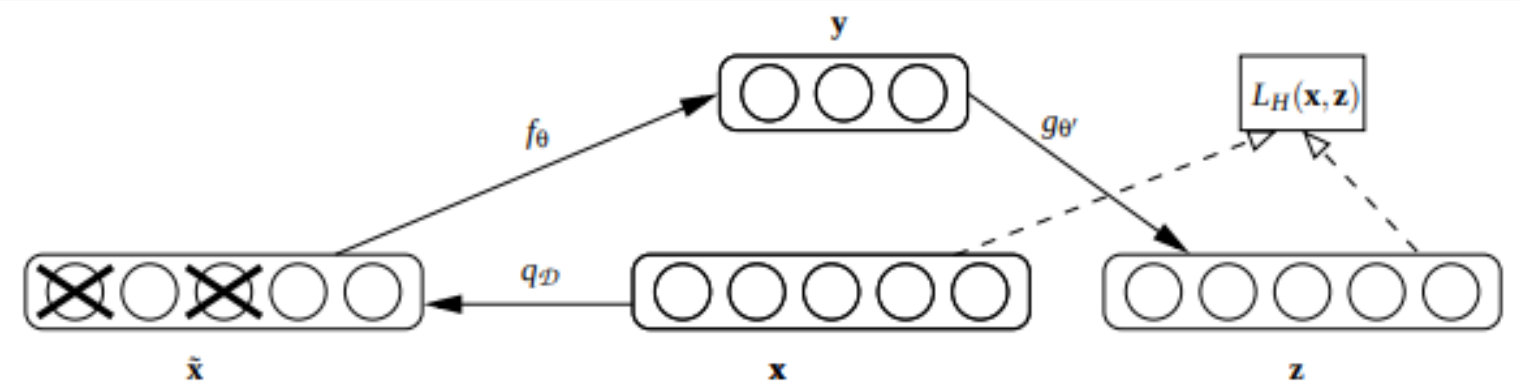
Training $\tilde{x}$ generated with :

- gaussian noise
- masking noise
- ...

Denoising autoencoders (DAE)

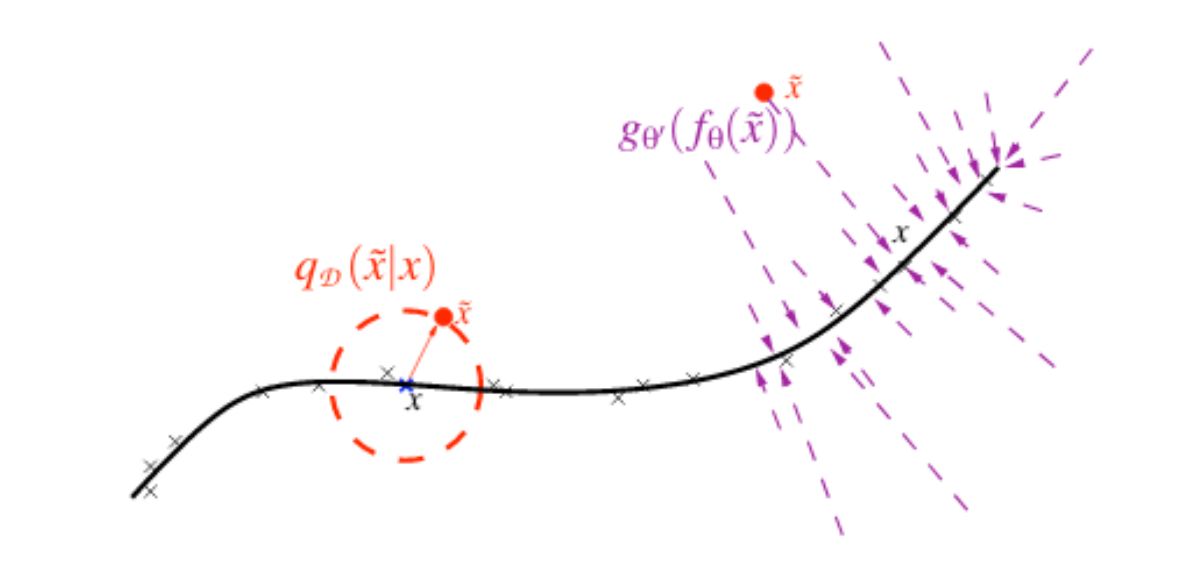
Recovering the values of the corrupted elements

Training is expected to capture dependencies



Denoising autoencoders (DAE)

The model learns with $p(x|\tilde{x})$ to project them back (via autoencoder $g'_\theta(f_\theta(\tilde{x}))$) onto the manifold.



Denoising autoencoders (DAE)

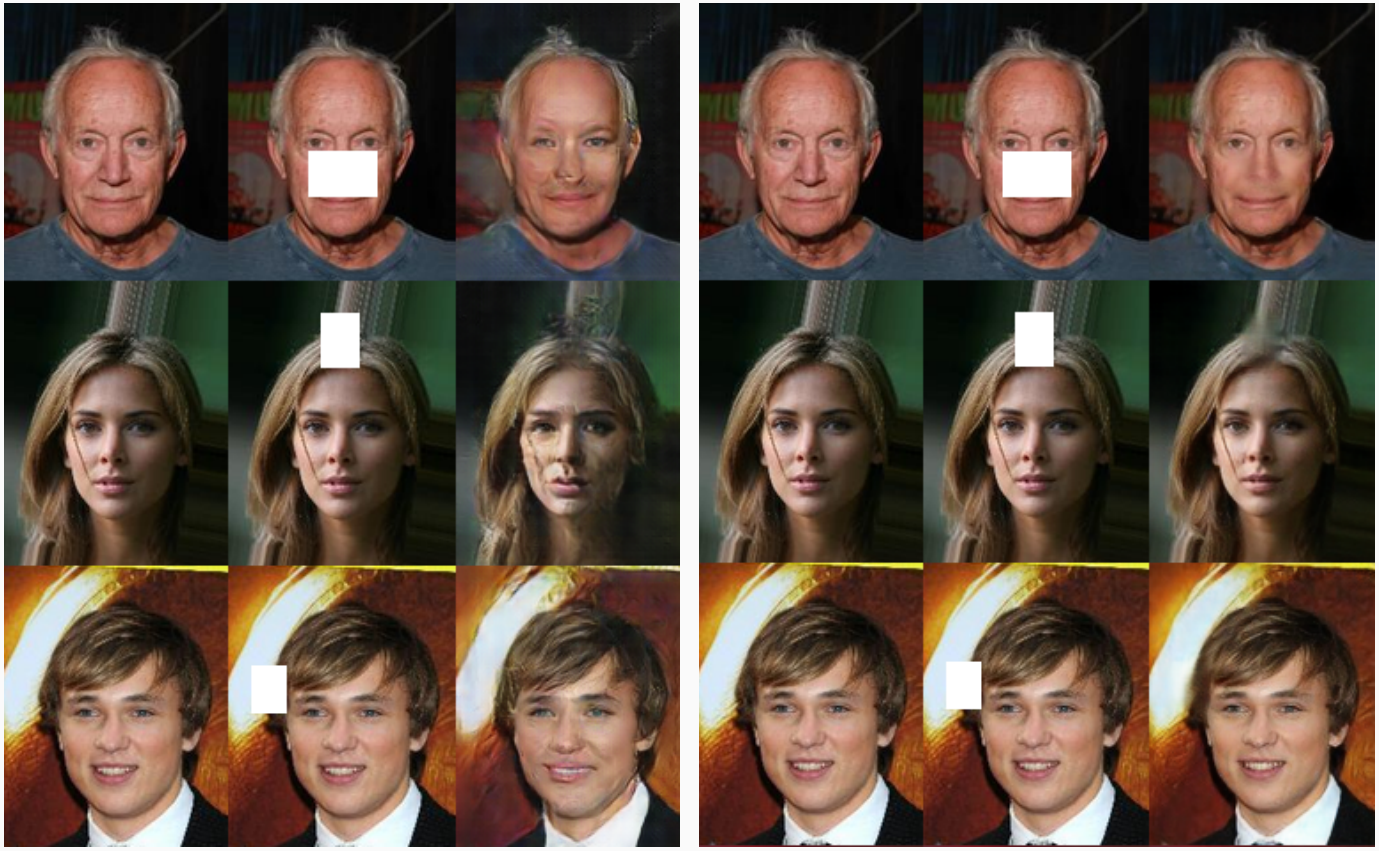
Emphasis : we give different weights to the input representation : α for the corrupted ones and β for the others. Thus we can make learning easier

$$L_{2,\alpha}(x, z) = \alpha\Big(\sum_{i \in \tau(\tilde{x})} (x_i - z_i)^2\Big) + \beta\Big(\sum_{i \notin \tau(\tilde{x})} (x_i - z_i)^2\Big) \qquad (4)$$

with $\tau(\tilde{x})$ the corrupted indexes

Denoising autoencoders (DAE)

Example of DAE for face completion :

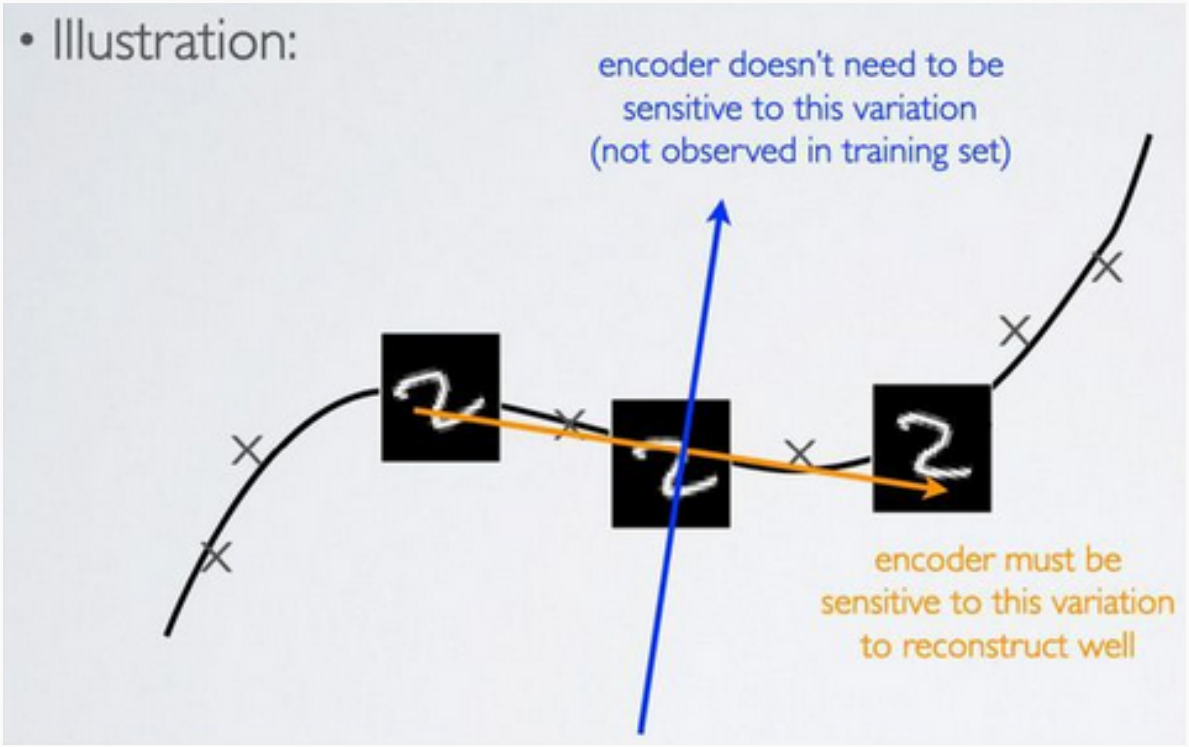


Contractive autoencoders

Contractive autoencoders

Similar to DAEs

Robust to slight variations of input values



Goal : extract only features that are variations of the training set

Contractive autoencoders

Needs to compute jacobian of hidden layers

$$J_{CAE}(\theta) = \sum_{x \in D_n} \left(\; L(x, g(f(x)) + \lambda ||J_f(x)||_F^2 \; \right) \tag{5}$$

Cost function of the traditional autoencoder and a regular constraint

Regularization using Frobenius norm and Jacobian matrix

Contractive autoencoders

Focus on representation robustness while DAE focuses on reconstruction

Harder to implement than DAE

Results :

- DAE stochastically
- CAE analytically : may be more stable

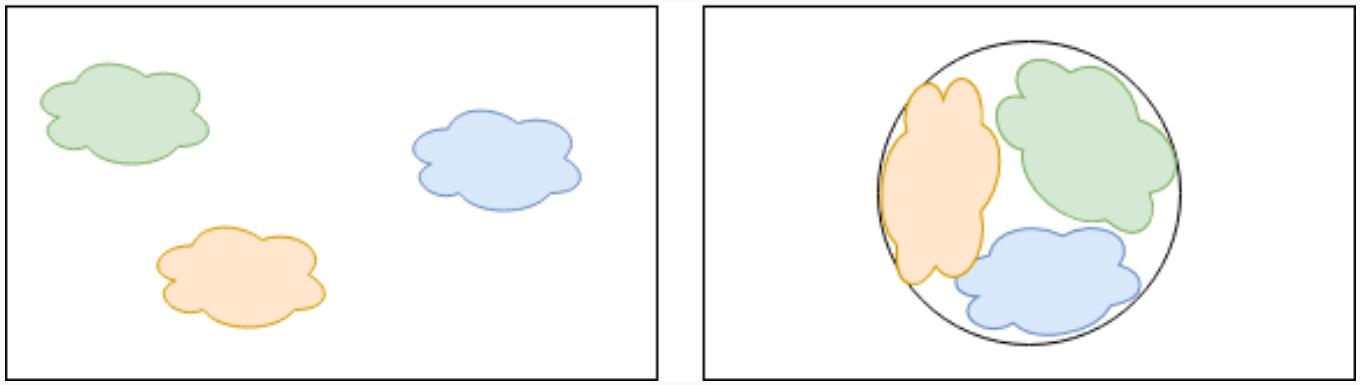
Variational autoencoders

Variational autoencoders

Learn the marginal likelihood, bayesian inference, force z to follow a gaussian distribution

Can generate data

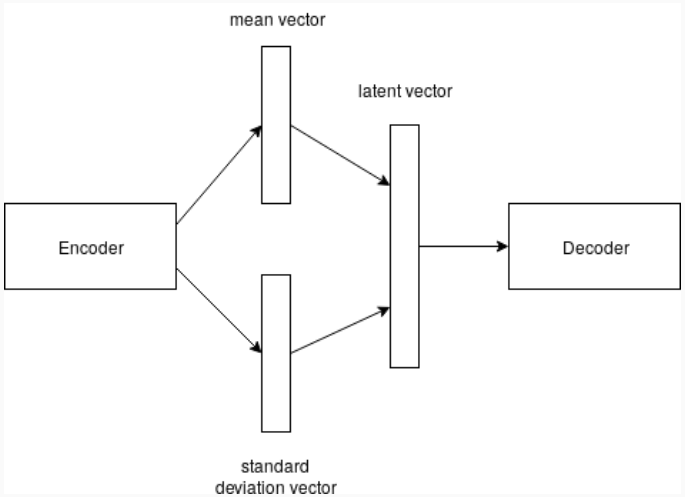
Without and with sampling :



Goal : maximize the variance (mean of 0, standard deviation of 1), avoid dead zones

Variational autoencoders

Instead of mapping any input to a fixed vector we want to map the input onto a distribution



Bottleneck vector is replaced by two separate vectors

Problem for backpropagation : we use a reparametrization trick

Variational autoencoders

Disentangled VAE :

- neurons are uncorrelated
- they learn something different

The probability distribution of the latent vector z

$L = RE + KL$ with RE : recognition error and KL : Kullback-Leitler (difference between a standard Normal (Gaussian) and the encoded distributiton

Higgins added a regularization parameter β

$L = RE + \beta \times KL$

Sources i

Ding, S., Zhang, J., Zhang, N., and Hou, Y. (2016).
Boltzmann Machine and its Applications in Image Recognition.
In *9th International Conference on Intelligent Information Processing (IIP)*, volume AICT-486 of *Intelligent Information Processing VIII*, pages 108–118, Melbourne, VIC, Australia.
Part 3: Deep Learning.

En, S., Crémilleux, B., and Jurie, F. (2017).
UNSUPERVISED DEEP HASHING WITH STACKED CONVOLUTIONAL AUTOENCODERS.
In *IEEE International Conference on Image Processing*, Bejing, China.

Sources ii


Hinton, G. and Salakhutdinov, R. (2006).
Reducing the dimensionality of data with neural networks.
Science (New York, N.Y.), 313:504–7.


Obin, N. and Beliao, J. (2018).
Sparse Coding of Pitch Contours with Deep Auto-Encoders.
In *Speech Prosody*, Poznan, Poland.

Sources iii

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011).
Contractive auto-encoders: Explicit invariance during feature extraction.
In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 833–840, USA. Omnipress.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010).
Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.

Sources iv

J. Mach. Learn. Res., 11:3371–3408.

Xu, Q., Wu, Z., Yang, Y., and Zhang, L. (2017). **The difference learning of hidden layer between autoencoder and variational autoencoder.** In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 4801–4804.