Deep Learning II : unsupervised tasks with Auto-encoders

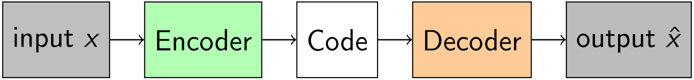
Moacir Ponti
ICMC, Universidade de São Paulo

Contact: `www.icmc.usp.br/~moacir` — `moacir@icmc.usp.br`

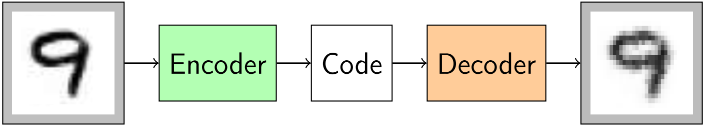
Teresina-PI/Brazil – Ago 2018

Agenda

General architecture of an Autoencoder



General architecture of an Autoencoder



Autoencoders basics: encoder and decoder

Encoder

Produces Code or Latent Representation

$$\mathbf{h} = s(\mathbf{W}\mathbf{x} + \mathbf{b}) = f(\mathbf{x})$$

Autoencoders basics: encoder and decoder

Encoder

Produces Code or Latent Representation

$$\mathbf{h} = s(\mathbf{W}\mathbf{x} + \mathbf{b}) = f(\mathbf{x})$$

Decoder

Produces Reconstruction of the input

$$\hat{\mathbf{x}} = s(\mathbf{W}'\mathbf{h} + \mathbf{b}') = g(\mathbf{h})$$

Tied weights when $\mathbf{W}' = \mathbf{W}^T$

Autoencoders basics: loss function

Given the output $\hat{\mathbf{x}} = g(f(\mathbf{x})$
We want to minimize some reconstruction loss:

$$\mathcal{L}(\mathbf{x}, g(f(\mathbf{x})) = \hat{\mathbf{x}})$$

Cross entropy (bits or probability vectors)

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \mathbf{x} \log \hat{\mathbf{x}} + (1 - \mathbf{x}) \log(1 - \hat{\mathbf{x}})$$

Mean squared error (continuous values)

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2$$

Autoencoders basics: flavours

Undercomplete

- Bottleneck layer produces code $\mathbf{h}$ with less dimensions then input $\mathbf{x}$

Overcomplete

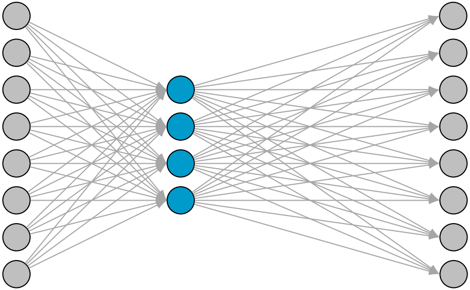
- Code $\mathbf{h}$ has more dimensions then the input $\mathbf{x}$
- Different versions e.g. sparse, denoising, contractive.

Agenda

Undercomplete

Learns a Lossy Compression of the input data.

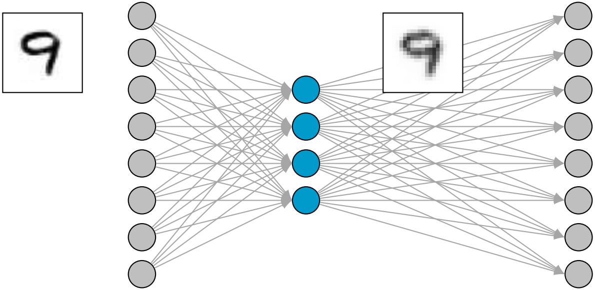
- has a “bottleneck” layer
- can be used for Dimensionality Reduction — often compared to Principal Component Analysis (PCA)
- often code is a good representation for the training data only



Undercomplete

Learns a Lossy Compression of the input data.

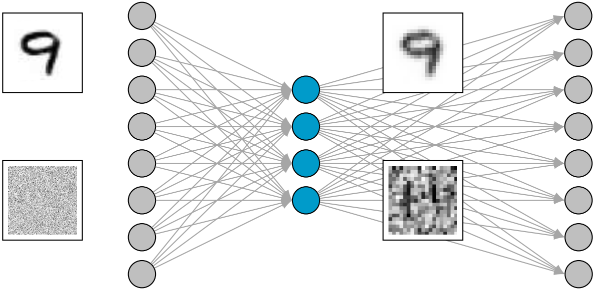
- has a “bottleneck” layer
- can be used for Dimensionality Reduction — often compared to Principal Component Analysis (PCA)
- often code is a good representation for the training data only



Undercomplete

Learns a Lossy Compression of the input data.

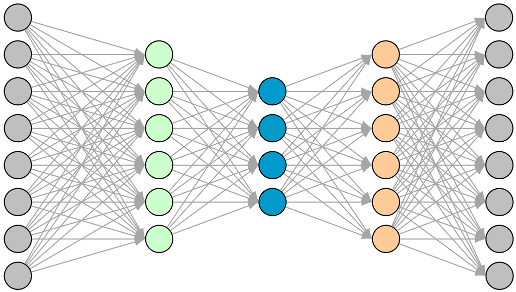
- has a "bottleneck" layer
- can be used for Dimensionality Reduction — often compared to Principal Component Analysis (PCA)
- often code is a good representation for the training data only



Undercomplete

Increasing the number of layers adds capacity to the AE.

- Encoder and Decoder layers can also be convolutional layers

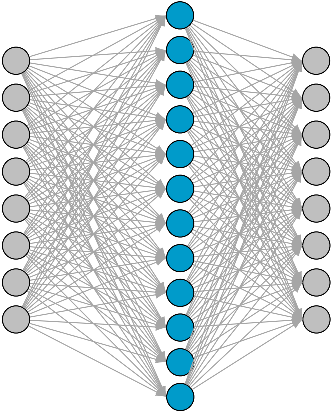


In principle with a sufficiently large capacity it may map every input to a single neuron on bottleneck layer.

Agenda

Overcomplete AEs

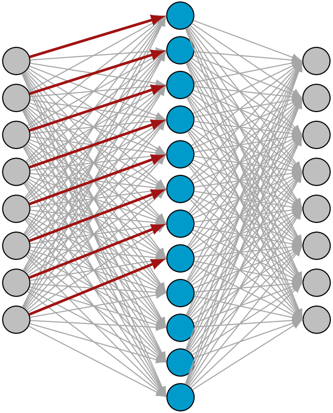
High-dimensional intermediate layer



Overcomplete AEs

High-dimensional intermediate layer

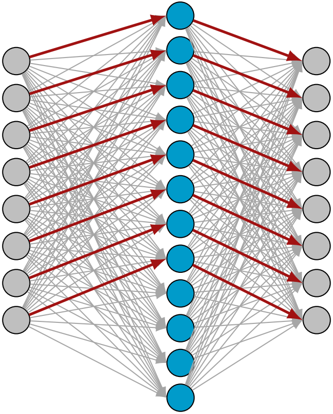
- a naive implementation would allow a copy so that $\mathbf{x} = \hat{\mathbf{x}}$



Overcomplete AEs

High-dimensional intermediate layer

- a naive implementation would allow a copy so that $\mathbf{x} = \hat{\mathbf{x}}$



Overcomplete regularized AEs

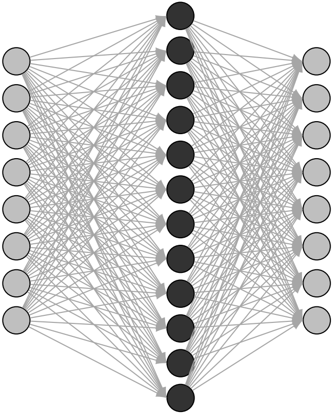
Regularization with sparsity constraint

$$\mathcal{L}(x, g(f(x))) + \Omega(f(x))$$
$$\mathcal{L}(x, g(f(x))) + \lambda \sum_i |h_i|,$$

- loss function tries to keep a low number of activation neurons per training input

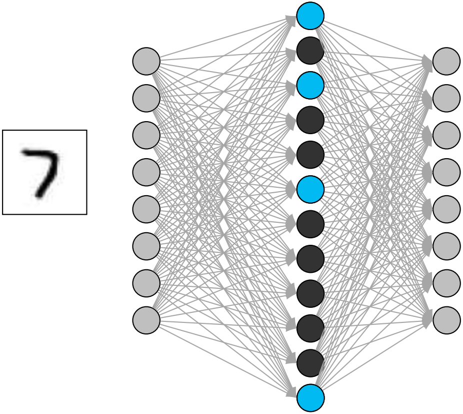
Overcomplete regularized AEs

Regularization with sparsity constraint



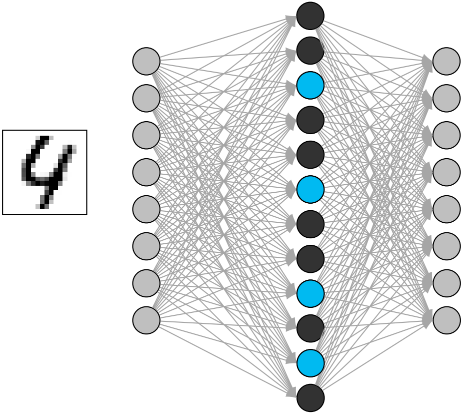
Overcomplete regularized AEs

Regularization with sparsity constraint



Overcomplete regularized AEs

Regularization with sparsity constraint



Denoising AEs (DAEs)

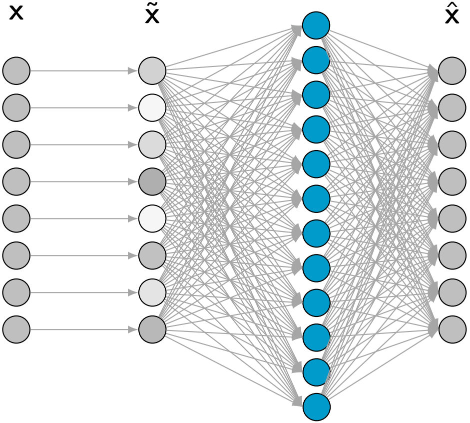
Regularization achieved by adding noise to $\mathbf{x}$

- the loss is computed using the noiseless input $\mathbf{x}$
- AE has to reconstruct $\mathbf{x}$ using a noisy input $\tilde{\mathbf{x}}$, so representation must be robust to noise
- this prevents the overcomplete AE to simply copy the data

Denoising AEs (DAEs)

Regularization achieved by adding noise to $\mathbf{x}$

- DAEs aim to learn a good internal representation as a side effect of learning to denoise the input



Denoising AEs (DAEs)

Noise processes

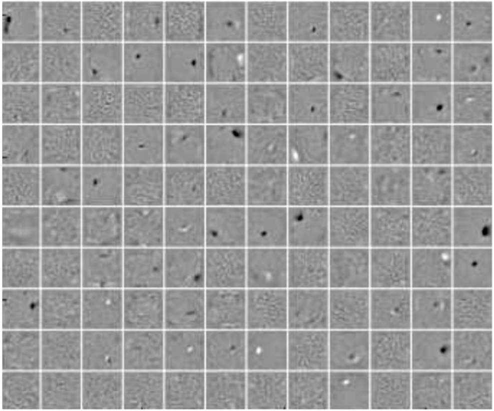
- Additive Gaussian Noise with $\mu = 0$, and some σ;
- Set a percentage of the input data to zero with some probability *p*.

Interpretation

- Learns to project data around some manifold to the distribution of the original (noiseless) data
- If some input is to far from the original distribution, it produces a high reconstruction error

Denoising AEs (DAEs): example

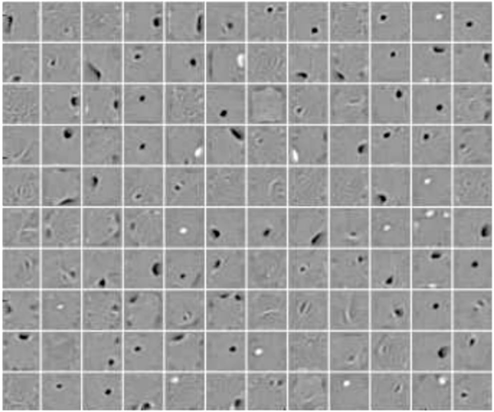
Using MNIST dataset, without noise



Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of Machine Learning Research, 2010: 3371-3408.

Denoising AEs (DAEs): example

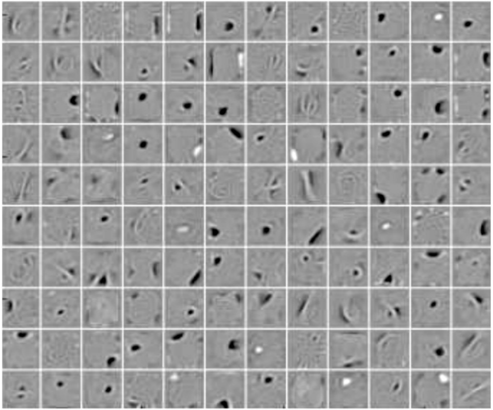
Using MNIST dataset, zero input variable with 25% probability



Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of Machine Learning Research, 2010: 3371-3408.

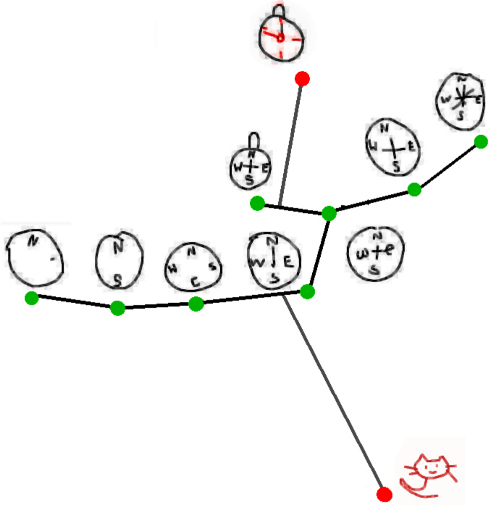
Denoising AEs (DAEs): example

Using MNIST dataset, zero input variable with 50% probability



Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of Machine Learning Research, 2010: 3371-3408.

A sketch manifold illustration



Concluding remarks

- AEs can be a good choice with unsupervised data;
- Deep autoencoders can be useful to many applications, via manifold learning;
- The potential for manifold learning can be used for instance on Generative tasks (Generative and Variational Autoencoders).
- Those can also be plugged in supervised architectures.

References

- **Ponti, M.; Paranhos da Costa, G. Como funciona o Deep Learning. Tópicos em Gerenciamento de Dados e Informações. 2017.**
- **Ponti, M.; Ribeiro, L.; Nazare, T.; Bui, T.; Collomosse, J. Everything you wanted to know about Deep Learning for Computer Vision but were afraid to ask. In: SIBGRAPI – Conference on Graphics, Patterns and Images, 2017.** `http://sibgrapi.sid.inpe.br/rep/sid.inpe.br/sibgrapi/2017/09.05.22.09`
- Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of Machine Learning Research, 2010: 3371-3408.
- Goodfellow, I., Bengio, Y., and Courville, A. Deep learning. MIT press, 2016.