From PCA to Autoencoders

Unsupervised Representation Learning

FLORENT FOREST

✉ forest@lipn.univ-paris13.fr
🌐 http://florentfo.rest
⭘ FlorentF9

17 décembre 2019

isae
Institut Supérieur de l'Aéronautique et de l'Espace
SUPAERO

Table of contents

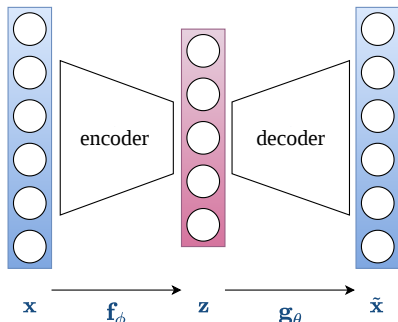# Introduction to autoencoders

[4]

Introduction to autoencoders

Definition

Definition

Definition

An autoencoder is a neural network trained to reconstruct its inputs. It is composed of two parts:

1. an encoder, mapping the input to a latent representation ("code") $\mathbf{z} = \mathbf{f}_\phi(\mathbf{x})$

2. a decoder, mapping the code back to the input space $\tilde{\mathbf{x}} = \mathbf{g}_\theta(\mathbf{z})$

Challenge

We do not want the encoder to learn the identity function, but to learn a *good representation* of our data.

Regularization?

Reducing the size of the *hypothesis set* $\mathcal{H}$ by constraining the space of possible solutions to the optimization problem.

- L2 weight decay
- Sparsity, L1 weight decay
- …

Introduction to autoencoders

Mathematical formulation

What is a good representation?

Let $q_{\phi}(Z|X)$ be a (stochastic) parametric mapping from X to Z. A good representation Z of a random variable X maximizes mutual information between X and Z (*infomax principle*):

$$\mathbb{I}(X; Z) = \mathbb{H}(X) - \mathbb{H}(X|Z)$$
$$= C(X) + \mathbb{E}_{q_{\phi}(X,Z)}[\log q_{\phi}(X|Z)]$$

For any parametric distribution $p_{\theta}(X|Z)$ we have
$\mathbb{E}_{q_{\phi}(X,Z)}[\log p_{\theta}(X|Z)] \leq \mathbb{E}_{q_{\phi}(X,Z)}[\log q_{\phi}(X|Z)]$ (using $D_{KL}(q||p) \geq 0$).

Task: maximize a lower bound on $\mathbb{I}(X; Z)$

$$\underset{\phi, \theta}{\text{maximize}}\ \mathbb{E}_{q_{\phi}(X,Z)}[\log p_{\theta}(X|Z)]$$

Loss function for deterministic autoencoders

We consider deterministic mappings $Z = \mathbf{f}_{\boldsymbol{\phi}}(X)$ (or $q_{\boldsymbol{\phi}}(Z|X) = \delta(Z - \mathbf{f}_{\boldsymbol{\phi}}(X))$) and $\tilde{X} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{f}_{\boldsymbol{\phi}}(X))$.

$$\underset{\boldsymbol{\phi}, \boldsymbol{\theta}}{\text{maximize}} \; \mathbb{E}_{q_{\boldsymbol{\phi}}(X)} \left[\log p_{\boldsymbol{\theta}}(X|Z = \mathbf{f}_{\boldsymbol{\phi}}(X)) \right]$$

Using empirical mean over a set of i.i.d. data samples:

$$\underset{\boldsymbol{\phi}, \boldsymbol{\theta}}{\text{maximize}} \; \sum_i \log p_{\boldsymbol{\theta}}(\mathbf{x}_i | \mathbf{z}_i = \mathbf{f}_{\boldsymbol{\phi}}(\mathbf{x}_i))$$

equivalent to:

$$\underset{\boldsymbol{\phi}, \boldsymbol{\theta}}{\text{maximize}} \; \sum_i \log p(\mathbf{x}^{(i)} | \tilde{\mathbf{x}}^{(i)} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{f}_{\boldsymbol{\phi}}(\mathbf{x}^{(i)})))$$

Let us turn this into a minimization the negative sum of individual loss functions $\mathcal{L}(\mathbf{x}|\tilde{\mathbf{x}}) = -\log p(\mathbf{x}|\tilde{\mathbf{x}})$.

Loss function for deterministic autoencoders

The reconstruction $\tilde{\mathbf{x}}$ is the mean of a distribution that may have generated $\mathbf{x}$.

Continuous variables: $\mathbf{x} \in \mathbb{R}^d$

- Gaussian distribution: $X|\tilde{X} = \tilde{\mathbf{x}} \sim \mathcal{N}(\tilde{\mathbf{x}}, \sigma^2 \mathbf{I})$
- Loss function: $\mathcal{L}(\mathbf{x}|\tilde{\mathbf{x}}) \propto ||\mathbf{x} - \tilde{\mathbf{x}}||_2^2$

$\rightarrow$ Mean Squared Error (MSE) loss

Binary variables: $\mathbf{x} \in \{0,1\}^d$, or $\mathbf{x} \in [0,1]^d$

- Bernoulli distribution: $X|\tilde{X} = \tilde{\mathbf{x}} \sim \mathcal{B}(\tilde{\mathbf{x}})$
- Loss function: $-\sum_{j=1}^{d}[\mathbf{x}_j \log \tilde{\mathbf{x}}_j + (1 - \mathbf{x}_j)\log(1 - \tilde{\mathbf{x}}_j)]$

$\rightarrow$ Cross-entropy loss

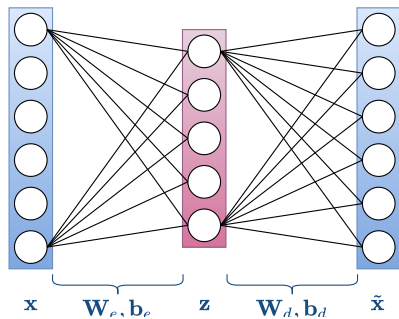# Neurons can learn principal components

[1], [6], [7]

Linear autoencoders

Show equivalence to PCA

[8]

Non-linear autoencoders

Non-linear autoencoders

Non-linear and deep autoencoders

Non-linear autoencoders

Non-linearity: sigmoid, tanh, ReLU…

Several layers

Not equivalent to PCA! (see paper AA-PCA)

Layer-wise pretraining: [4] In fact, not necessary (source?).

Non-linear autoencoders

Different types of regularization

Problem: even a single continuous latent variable can remember the entire training set (one real number per sample).

The latent space is not continuous.

Let's code!

[9]

Non-linear autoencoders

Variational autoencoders (VAE)

Limitation of standard AE: the latent space has *no structure* and may not be continuous; we cannot explore it nor sample from it.

Probabilistic setting

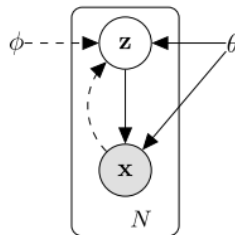Generative model $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$

1. $\mathbf{z}$ sampled from $p_{\boldsymbol{\theta}}(\mathbf{z})$ (prior)

2. $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ (likelihood/*probabilistic decoder*)

Recognition model:
$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ (approximate posterior/*probabilistic encoder*)

Stochastic gradient variational Bayes (SGVB) [5] is an efficient method to estimate the parameters in case of intractable likelihood/posterior and large datasets (as in DL).

VAE loss function

VAE ELBO (evidence lower bound)

$$\underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\text{maximize}} \ - D_{KL}\left(q_{\boldsymbol{\phi}}(Z|X)||p_{\boldsymbol{\theta}}(Z)\right) + \mathbb{E}_{q_{\boldsymbol{\phi}}(Z|X)}\left[\log p_{\boldsymbol{\theta}}(X|Z)\right]$$

Key ideas

- The second term is a (negative) reconstruction error (e.g. MSE or cross-entropy) as in a deterministic AE.
- The first term, a Kullback-Leibler divergence between $q_{\boldsymbol{\phi}}(Z|X)$ and $p_{\boldsymbol{\theta}}(Z)$, acts as a regularizer pushing the encoder distribution closer to the prior distribution (typically a gaussian)

VAE loss function

Let's put gaussians everywhere!

- $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$
- $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$

The reparameterization trick

To sample from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, we use the reparameterization
$\mathbf{z} = f_{\boldsymbol{\phi}}(\mathbf{x}, \epsilon) = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

For a given $\mathbf{x}^{(i)}$, and using 1-sample Mont-Carlo estimation, the ELBO becomes:

$$\frac{1}{2} \sum_j \left(1 + \log(\boldsymbol{\sigma}_j^{(i)})^2 - \boldsymbol{\mu}_j^{(i)} - (\boldsymbol{\sigma}_j^{(i)})^2 \right) + \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i)})$$

$$\text{where } \mathbf{z}^{(i)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \cdot \boldsymbol{\epsilon}$$

Let's code!

Applications

Dimensionality reduction, useful features

Pre-processing pour d'autres algos (regress, classif, clustering)

Unsupervised pretraining -> supervised finetuning

[3]

Dimensionality reduction Data-specific, lossy compression.

The decoder model can be used to generate new data samples:

- Deterministic AEs: adding noise, interpolating or extrapolating in latent space [2]
- Generative models (VAE, GAN)

When the input is different from the training data distribution (e.g. an outlier), the autoencoder will produce a large reconstruction error. This error can be used to score anomalies [?].
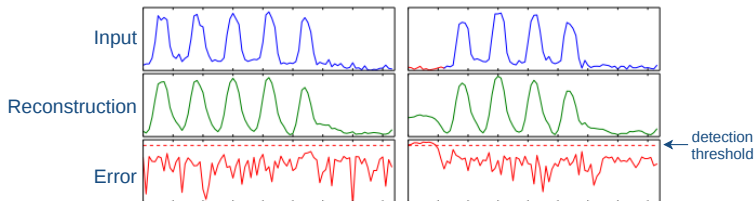
(Malhotra et al, 2016)

📄 S. Becker.
Unsupervised Learning Procedures for Neural Networks.
The International Journal of Neural Systems, 1:17–33, 1991.

📄 T. Devries and G. W. Taylor.
Dataset Augmentation in Feature Space.
In *ICLR 2017 Workshop*, pages 1–12, 2017.

📄 D. Erhan, A. Courville, and P. Vincent.
Why Does Unsupervised Pre-training Help Deep Learning ?

Journal of Machine Learning Research, 11:625–660, 2010.

G. E. Hinton and R. Salakhutdinov.
Reducing the Dimensionality of Data with Neural Networks.
Science, 313(July):504–507, 2006.

D. P. Kingma and M. Welling.
Stochastic Gradient VB and the Variational Auto-Encoder.
2nd International Conference on Learning Representations (ICLR), pages 1–14, 2014.

E. Oja.
A Simplified Neuron Model as a Principal Component Analyzer.
Journal of Mathematical Biology, 15(3):267–273, 1982.

📄 E. Oja.
Principal Components, Minor Components, and Linear
Neural Networks, 1992.

📄 E. Plaut.
From Principal Subspaces to Principal Components with
Linear Autoencoders.
pages 1–6, 2018.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol.
Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.
Journal of Machine Learning Research, 11:3371–3408, 2010.