Deep learning : focus on autoencoders as a pre-processing step for classification

Julien Wojak - Institut Fresnel

2020-01-11

Table of contents

Introduction

- Supervised Classification : give label to unlabelled data, learn how to label with examples
- Unsupervised Classification or Clustering : goup similar data

- Supervised Classification : give label to unlabelled data, learn how to label with examples
- Unsupervised Classification or Clustering : goup similar data

Examples

- Supervised Classification : give label to unlabelled data, learn how to label with examples
- Unsupervised Classification or Clustering : goup similar data

Examples

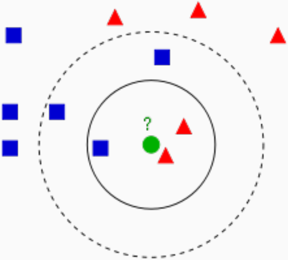
- E-mail classification: *spam* or *ham*

- Supervised Classification : give label to unlabelled data, learn how to label with examples
- Unsupervised Classification or Clustering : goup similar data

Examples

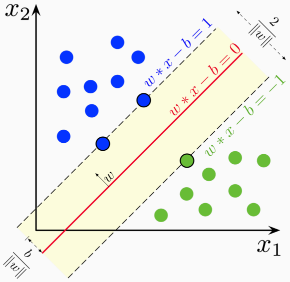
- E-mail classification: *spam* or *ham*
- CME classification
- detection of groups of similar website visitors

- k-Nearest Neighbors



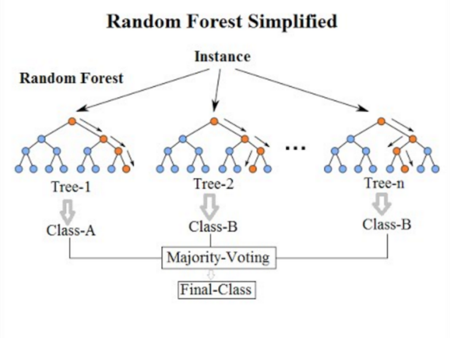
Non exhaustiv list of classification methods

- k-Nearest Neighbors
- Support Vector Machine

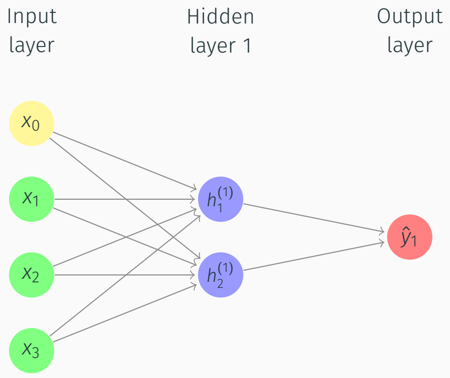


Non exhaustiv list of classification methods

- k-Nearest Neighbors
- Support Vector Machine
- Decision Tree and Random Forest



- k-Nearest Neighbors
- Support Vector Machine
- Decision Tree and Random Forest
- Deep Learning (Neural networks)



Curse of dimensionality // overfitting

- high dimension feature space => enormous amount of training data
- overfitting // poor generalization: neural network learn ”by heart”

Curse of dimensionality // overfitting

- high dimension feature space => enormous amount of training data
- overfitting // poor generalization: neural network learn ”by heart”
- Need of dimensionality reduction tools

- high dimension feature space => enormous amount of training data
- overfitting // poor generalization: neural network learn ”by heart”

- Need of dimensionality reduction tools
    - feature selection: selection of a subset of relevant features

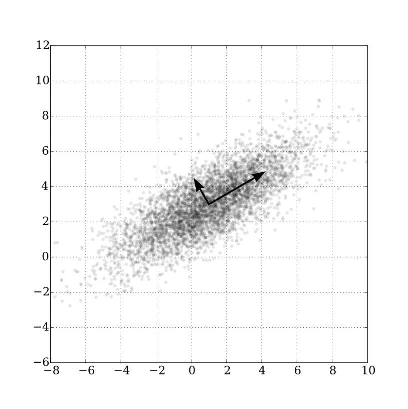
Curse of dimensionality // overfitting

- high dimension feature space => enormous amount of training data
- overfitting // poor generalization: neural network learn ”by heart”
- Need of dimensionality reduction tools
    - feature selection: selection of a subset of relevant features
    - feature extraction: build derived informative and non-redondand features

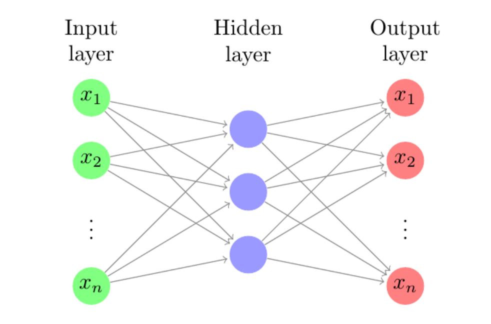
Dimentionality reduction

PCA

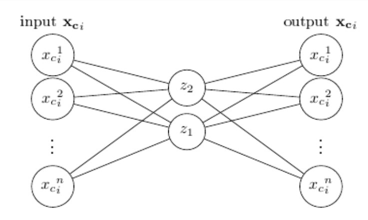
- PCA Principal Component Analysis
- main idea: find projection on the hyperplane that lies closest to the data
- preserve the maximum amount of variance
- minimize the mean square error between original datasests and projected datasets

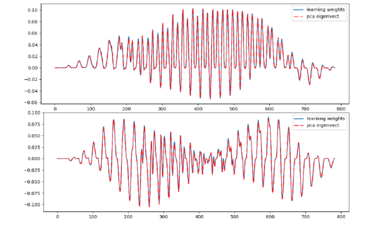


- Autoencoders (auto-associative neural network) : use to learn efficient representation of the input data
- learn the identity: copy inputs to outputs, minimize MSE between inputs and output



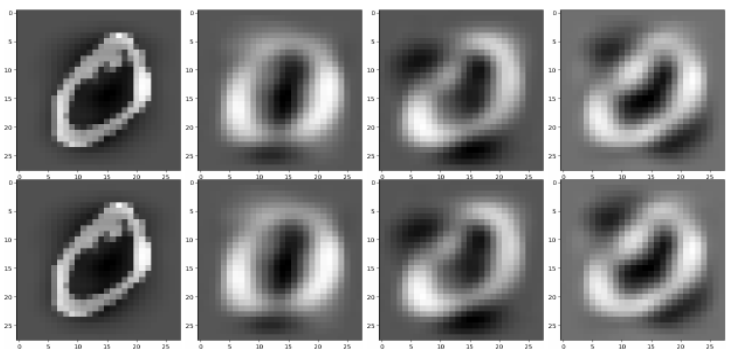
- using tie Autoencoder, with MSE and with linear activations one can see that PCA and Autoencoders are equivalent

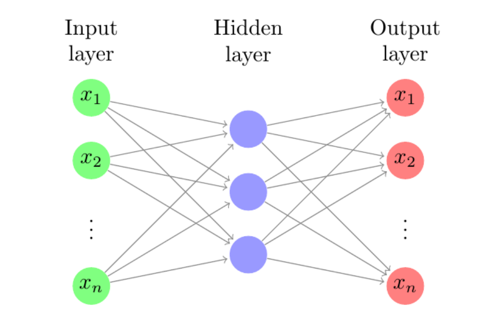




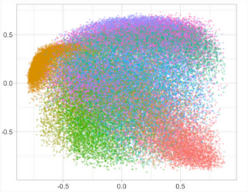
PCA and autoencoder: Reconstruction

- reconstruction from 784 dimensions to 1, 2 and 10 dimensions

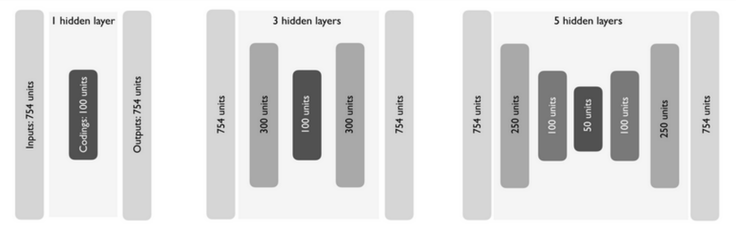




- hidden layer if smaller than input and output layers
- the network needs to learn a lossy compression of datasets
- possibility to add prior (weight regularisation)



- stack several hidden layers
- represention of more complex relationship for compression
- learn sequentially each hidden layer, group all layers for fine tuning



Examples in Hyperspectral imaging

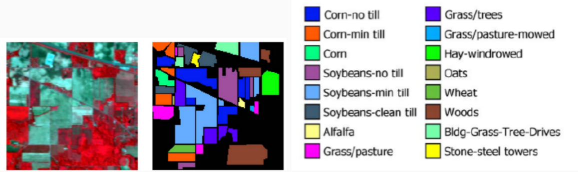
- Stack of images
- Each image = narrow wavelength range of electromagnetic spectrum

Hyperspectral imaging

- Stack of images
- Each image = narrow wavelength range of electromagnetic spectrum

Example:

- AVARIS Sensor
- 145x145 pixels images
- 224 bands (0.4 to $2.5 * 10^{-6}$ meters)
- agriculture, forest, natural vegetation



Example 1

Spectral-Spatial Classification of Hyperspectral Image Using Autoencoders

Zhouhan Lin, Yushi Chen, Xing Zhao Gang Wang

***Abstract*—Hyperspectral image (HSI) classification is a hot topic in the remote sensing community. This paper proposes a new framework of spectral-spatial feature extraction for HSI classification, in which for the first time the concept of deep learning is introduced. Specifically, the model of autoencoder is exploited in our framework to extract various kinds of features. First we verify the eligibility of autoencoder by following classical spectral information based classification and use autoencoders with different depth to classify hyperspectral image. Further in the proposed framework, we combine PCA on spectral dimension and autoencoder on the other two spatial dimensions to extract spectral-spatial information for classification. The experimental results show that this framework achieves the highest classification accuracy among all methods, and outperforms classical classifiers such as SVM and PCA-based SVM.**

Keywords-autoencoders; deep learning; hyperspectral; image classification; neural networks; stacked autoencoders

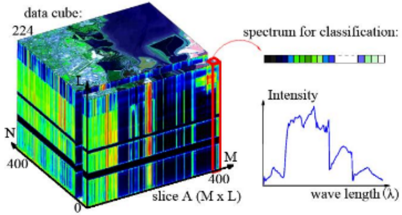
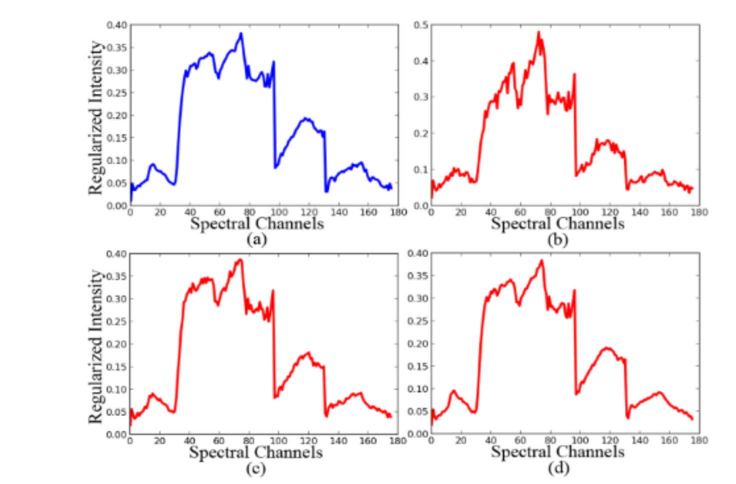
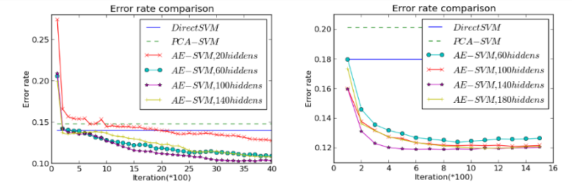


Figure 1. A typical scene of hyperspectral image. Each pixel consists of a whole spectrum.

Reconstruction with 1, 100, 1000 epochs



Classification results (KSC and Pavia datasets)



Example 2

Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging

Jaime Zabalza [a], Jinchang Ren [a,*], Jiangbin Zheng [b], Huimin Zhao [c], Chunmei Qing [d], Zhijing Yang [e], Peijun Du [f], Stephen Marshall [a]

A B S T R A C T

Stacked autoencoders (SAEs), as part of the deep learning (DL) framework, have been recently proposed for feature extraction in hyperspectral remote sensing. With the help of hidden nodes in deep layers, a high-level abstraction is achieved for data reduction whilst maintaining the key information of the data. As hidden nodes in SAEs have to deal simultaneously with hundreds of features from hypercubes as inputs, this increases the complexity of the process and leads to limited abstraction and performance. As such, segmented SAE (S-SAE) is proposed by confronting the original features into smaller data segments, which are separately processed by different smaller SAEs. This has resulted in reduced complexity but improved efficacy of data abstraction and accuracy of data classification.

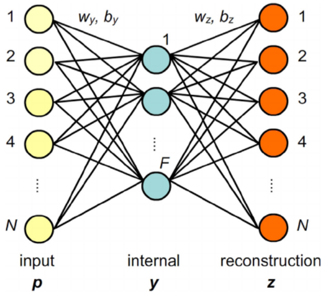


Fig. 1. Basic AE scheme.

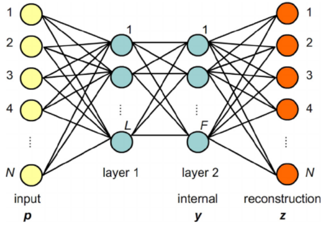


Fig. 2. Stacked AE scheme (2 layers).

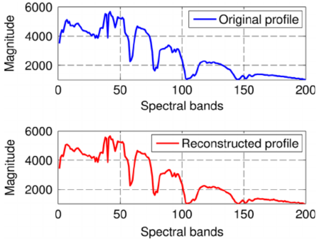


Fig. 3. Original spectral profile (top) and reconstructed one by SAE (bottom).

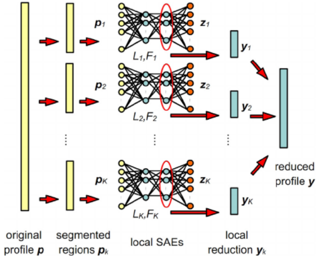
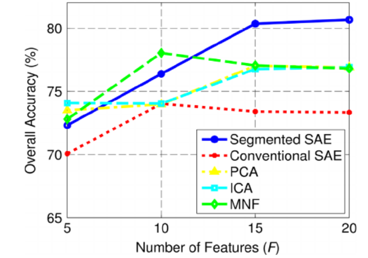
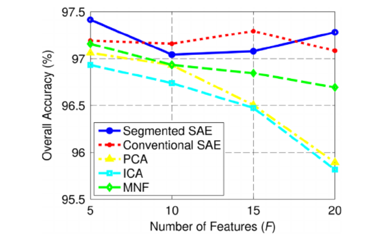


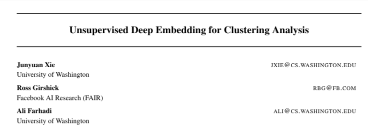
Fig. 4. S-SAE structure using several two-layer SAEs.


Overall Accuracy (%)
80
75
70
65
5
10
15
20
Number of Features (F)
Segmented SAE
Conventional SAE
PCA
ICA
MNF


Overall Accuracy (%)
97.5
97
96.5
96
95.5
5
10
15
20
Number of Features (F)
Segmented SAE
Conventional SAE
PCA
ICA
MNF

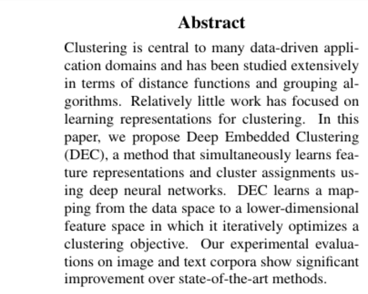
Conclusion

- Autoencoders are powerful tools to perform dimensionality reduction
- Improvements of classification results using AE
- First step towards a full non supervised classification



Unsupervised Deep Embedding for Clustering Analysis

Junyuan Xie JXIE@CS.WASHINGTON.EDU
University of Washington
Ross Girshick RBG@FB.COM
Facebook AI Research (FAIR)
Ali Farhadi ALI@CS.WASHINGTON.EDU
University of Washington



Abstract

Clustering is central to many data-driven application domains and has been studied extensively in terms of distance functions and grouping algorithms. Relatively little work has focused on learning representations for clustering. In this paper, we propose Deep Embedded Clustering (DEC), a method that simultaneously learns feature representations and cluster assignments using deep neural networks. DEC learns a mapping from the data space to a lower-dimensional feature space in which it iteratively optimizes a clustering objective. Our experimental evaluations on image and text corpora show significant improvement over state-of-the-art methods.

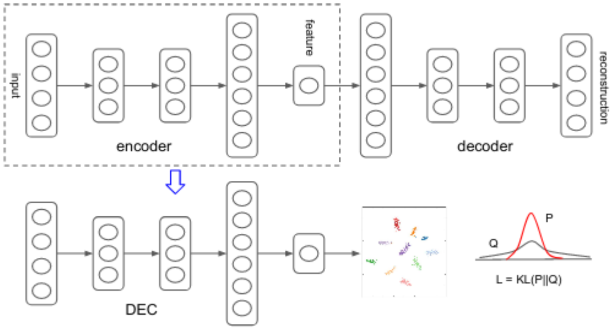
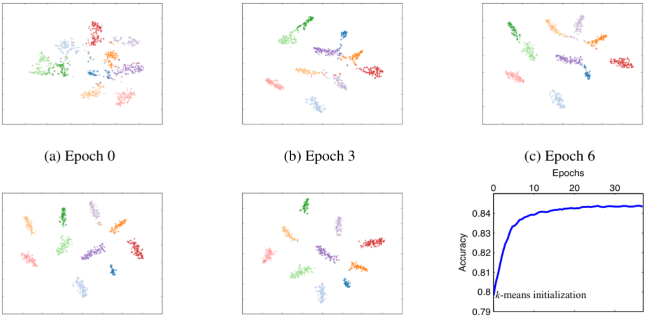
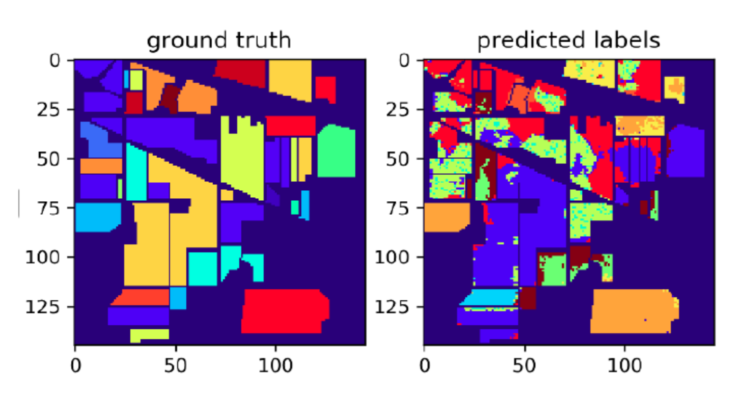


Figure 1. Network structure



(a) Epoch 0 (b) Epoch 3 (c) Epoch 6

(d) Epoch 9 (e) Epoch 12 (f) Accuracy vs. epochs


ground truth
predicted labels
0
25
50
75
100
125
0
50
100