Machine Learning Engineer Nanodegree

Capstone Proposal

Valérian Wrobel

march 13th, 2017

Proposal

Domain Background

Currently, computer vision is one of the most prominent application of machine learning and encompasses various tasks from digit classification to face recognition. In recent years, various Deep Neural Networks architectures have been studied for supervised learning as well as for unsupervised learning applied to images. Since raw images are high dimensional datapoints with strong redundancy, performant feature (or representation) learning that allow dimensionality reduction is a key aspect for any computer vision problem. In this capstone project, I would like to focus on a deep neural network architecture fitted for unsupervised learning problems.

Generative adversarial networks (GAN) have been introduced by *Ian Goodfellow et al.* in 2014 and have since been used in a wide variety of unsupervised machine learning applications to computer vision. For example, in this article published in 2016, *Alec Radford et al.* describe a method for feature learning based on Deep convolutional generative adversarial networks. These methods do not aim specifically at the clustering of datapoints during feature learning. In this article entitled Unsupervised Deep Embedding for Clustering Analysis published in 2016, *Junyuan Xie et al.* present a model architecture - according to their own words - "that simultaneously learns feature representations and cluster assignments using deep neural networks".

To evaluate the potential of a photovoltaic installation on an individual house, knowing the type of the roof as well as its inclination and orientation is crucial. A method that allows the identification of roof types for large batches of houses can be very valuable. Satellite or aerial images used in conjonction with building footprints can provide pictures of the roofs in a given neighbourhood. I would like to test the *Junyuan Xie et al.* model for roof images segmentation.

Problem Statement

For unlabeled images, finding a relevant segmentation of roofs requires efficient feature learning and clustering. Measuring performance of an unsupervised learning model can be tricky since there is no ground truth if the dataset is unlabeled.

We propose two ways to measure the performance of the model and of the benchmark:

- first test the two models on the MNIST dataset where the data is labeled and where we know the number of expected clusters (10),
- then adapt the models to the roofs dataset and measure the silhouette score of each clustering (for several numbers of clusters).

We considered using the datasets created by the author of this thesis on Machine Learning for Aerial Image Labeling but it seemed harder to handle than MNIST for a first approach of the models.

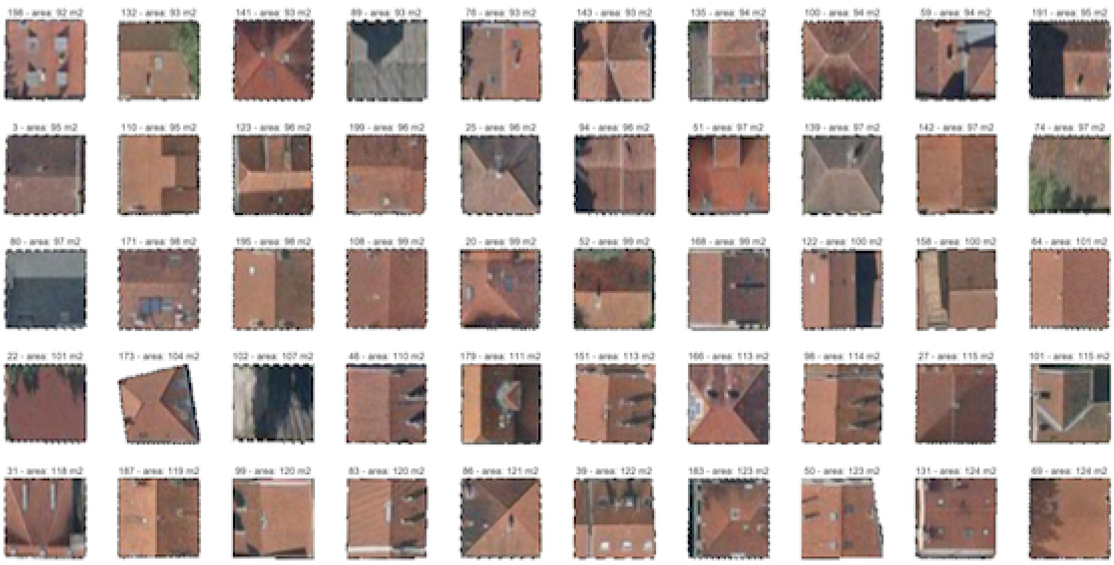
The visual representations of the results will help to know if the model can help for roofs segmentation in the context of photovoltaic studies.

For reference, in their 2015 article entitled Learning Classifiers from Synthetic Data Using a Multichannel Autoencoder, *Xi Zhang et al.* aim at clustering a roof dataset. They distinguish six kinds of roofs: flat, gable, gambrel, half-hip, hip, pyramid.

Datasets and Inputs

The metropole of Lyon in France promotes open data sharing. In that context, aerial images of the city are made available from a web map service. Moreover, Open Stret Map provides an open access api to the buildings footprints of a given area as xml data. Combining both data sources, we can get roof images for the buildings of a given area.

As we can see on the picture above, the resulting dataset will not be perfectly clean because the footprints do not correspond exactly to the houses on the aerial images. Moreover some shadows casted on the roofs can corrupt some datapoints. Furthermore footprints can be polygons with various shapes and sizes. As a first approach to roof segmentation, I propose to consider only rectangular footprints and to build a dataset of square images by stretching all raw images to the same size. The resulting dataset would look like the following sample.



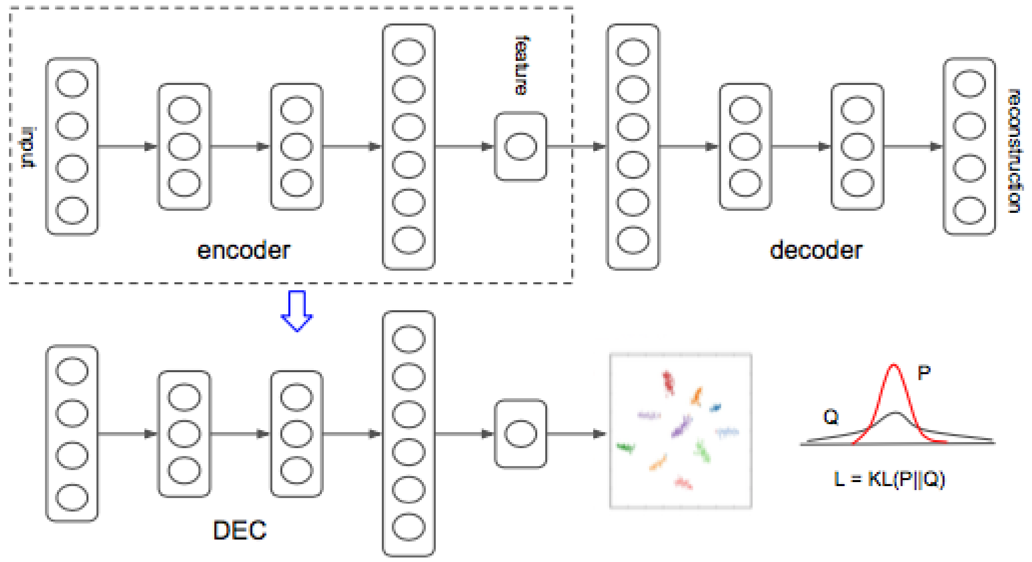
Given the available aerial images, we could build a set of 20000 roof images.

Solution Statement

Junyuan Xie et al. describe their DEC model thoroughly in their article. The DEC model is trained in two phases:

1. parameter initialization with a deep autoencoder,
2. parameter optimization (i.e., clustering) with computing of an auxiliary target distribution and minimizing the Kullback–Leibler divergence to it.

The following figure taken from the article illustrates the network structure they propose:



As a starting point, I propose to use [a Keras implementation]() of the DEC model available on github.

Junyuan Xie et al. have tested their model on several classic datasets: they claim an accuracy of 84.30% on clustering the MNIST dataset. I propose to use their model to tackle the roofs dataset.

Benchmark Model

Principal Component Analysis (PCA) for dimensionality reduction combined with k-means could be a benchmark model for our problem. Instead, we propose to compare our results to a model that combines t-distributed stochastic neighbor embedding (t-SNE) for dimensionality reduction and k-means for clustering. As suggested in the scikit-learn documentation, we might have to run a PCA first to reduce the number of dimensions "to a reasonable amount" (e.g. 50) and then to apply t-SNE in order to allow non linear transformations of the reduced data.

t-SNE is a very popular machine learning algorithm for dimensionality reduction of data developed by Geoffrey Hinton and Laurens van der Maaten. For the MNIST dataset for exemple, this algorithm allows an efficient segmentation (as can be seen in the middle of [this webpage]()).

I propose to start building a DEC aiming the segmentation of the MNIST dataset and compare its performance to t-SNE with k-means. The MNIST dataset is a good starting point since we know the number of clusters we are looking for (10) and the label of each datapoint. Hence we could measure the accuracy of clustering for each model. After this first step, we can adapt the DEC model to the roofs dataset and compare it again to t-SNE followed by k-means.

Evaluation Metrics

Since we propose to first train test our DEC and benchmark models on the MNIST dataset, we can use the available labels and calculate the unsupervised learning accuracy defined as:

$$ACC = \max_m \frac{\sum_{i=1}^{n} \mathbf{1}\{l_i = m(c_i)\}}{n}$$

where l_i is the ground-truth label, c_i is the cluster assignment produced by the algorithm, and m ranges over all possible one-to-one mappings between clusters and labels.

Moreover, we propose to calculate the sihouette metric for both models and different values for the numbers of clusters.

The silhouette value measures how an object is similar to the cluster it is assigned to compared to other clusters. Let us consider a dataset segmented in k clusters. We call C the set of clusters. We define the dissimiliarity of a datapoint i to any cluster c, $d_c(i)$, as the average of the distance from i to all points in c. We will consider the euclidian distance for the dissimilarity calculation. The silhouette score of a datapoint i assigned to the cluster c is:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

where:

- $a(i) = d_c(i)$, is the dissimilarity of i to its cluster,
- $b(i) = \min_{c' \in C, c' \neq c} d_{c'}(i)$ is the lowest average dissimilarity of i to any other cluster.

From the above definition, we have $-1 \leq s(i) \leq 1$. If the silhouette value is close to 1, it means that $a(i) \ll b(i)$ and so that the clustering is clear for the datapoint i. $s(i)$ close to 0 means that the datapoint is at the border of two clusters. $s(i)$ close to -1 means that the clustering is inapropriate for this point.

Hence averaging the silhouette score over all datapoints for a given clustering model and a given number of clusters is a way to measure the quality of the clustering without needing ground truth labels.

For the roofs dataset, we do not have any ground truth labels: we can only compare our models with the sihouette score.

Project Design

Roofs dataset preparation

The first task we will perform is to prepare the dataset of roofs. We will proceed with the following steps:

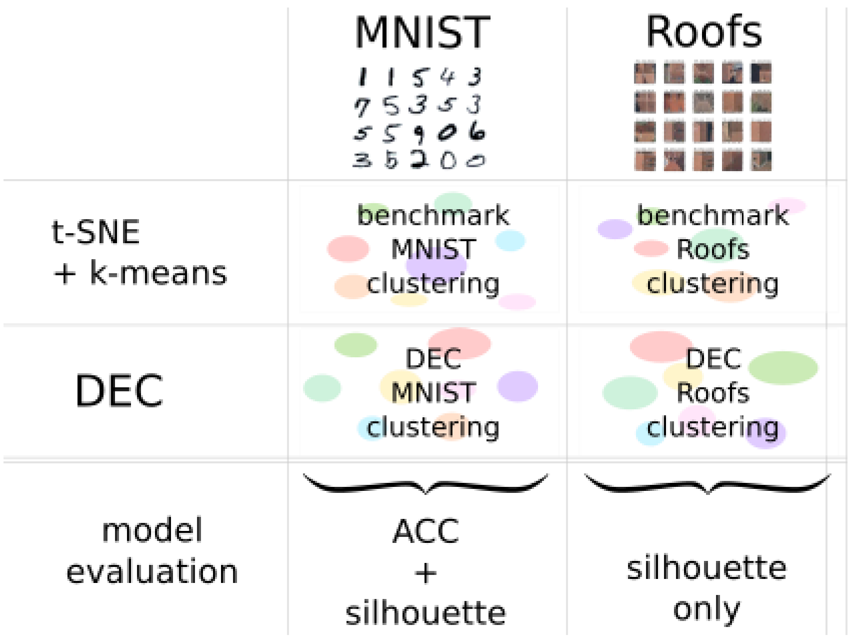
1. select an area of the Lyon metropole for which we get aerial images and building footprints xml data,
2. import the footprints as geopandas objects in Python,
3. filter the geopandas polygons to retain only rectangles within a given surface range (polygons of 4 points between 50m2 and 200m2) in order to keep only individual houses that can easily fit in a square image,
4. crop the aerial image with the selected rectangles,
5. rotate and stretch the cropped images so as to have square roofs images,
6. transform the images to get them centered on 0 with values in [-1, 1].

During this dataset preparation, we will have to study the distribution of areas and of polygon nodes counts for all the footprints retrieved from Open Street Map.

Clustering models

I intend to proceed the following way:

1. implement the benchmark model with t-SNE and k-means scikit-learn functions,
2. apply the clustering with this model on the MNIST dataset,
3. evaluate the model performance with ACC and silhouette scores,
4. implement a DEC network starting from the Keras implementation available on github,
5. apply the clustering with this model on the MNIST dataset,
6. evaluate the model performance with ACC and silhouette scores,
7. develop relevant visualisations of the clustered dataset,
8. adjust the benchmark model to the roofs dataset,
9. apply the clustering with the adjusted model on the roofs dataset,
10. evaluate the model performance with silhouette scores,
11. adjust the DEC model to the roofs dataset,
12. apply the clustering with this model on the roofs dataset,
13. evaluate the model performance with silhouette scores,
14. develop relevant visualisations of the clustered dataset.



Then we could draw some conclusions on the performance of the DEC model to cluster roofs images by type compared to our t-SNE benchmark. The main questions we intend to answer are: Is DEC a usefull model to help finding the photovoltaic potential of each house in a given area? Does it perform better than t-SNE followed by k-means ?