Mini-project 1
Kohonen maps on hand-written digits

Rodrigo C. G. Pena
rodrigo.pena@epfl.ch

November 29, 2016

Abstract

This report describes an implementation of Kohonen maps, also known as Self-Organizing Maps (SOM), and their application on a dataset of hand-written digits.

1 Introduction

Kohonen map (or self-organizing map) is an algorithm for unsupervised learning of a lower-dimensional representation of data. It was proposed originally in Kohonen (1982). The representation learned by Kohonen's algorithm consists of a square grid of neurons (henceforth called prototypes), whose vertices' coordinates are learned such that the grid is embedded on the hypersurface on which lays the input data cloud.

Let $D \in \mathbb{R}^{n \times m}$ be our data matrix, constructed by stacking of n samples of the m-dimensional variables of interest. Denote by $\mathbf{d}_i \in \mathbb{R}^m, i \in \{1, \ldots, n\}$ each row (datapoint) of this matrix. Also, denote by $\mathbf{w}_v \in \mathbb{R}^m$ the coordinates of prototype v, which are initially at a random position. At each step of Kohonen's algorithm, a single datapoint is presented, and we find the best matching unit (b.m.u.) to it in the current map:

$$\text{b.m.u.} = \arg\min_v \|\mathbf{w}_v - \mathbf{d}_i\|_2^2. \tag{1}$$

Then, each of the prototypes are updated according to the following rule:

$$\mathbf{w}_v \leftarrow \mathbf{w}_v + \eta K_\sigma(v)(\mathbf{d}_i - \mathbf{w}_v), \tag{2}$$

where η is a learning rate, and $K_\sigma : \mathbb{Z}^2 \to \mathbb{R}$ is a neighborhood discount function, centered at the b.m.u., and given by

$$K_\sigma(v) = \exp\left[\frac{d^2(\text{b.m.u.}, v)}{2\sigma^2}\right]. \tag{3}$$

We call σ the neighborhood width, and $d : \mathbb{Z}^2 \to \mathbb{R}$ is the Euclidean distance of two points on the map expressed in *grid coordinates*. Update rule 1 is repeated until convergence of the coordinates.

It is convenient sometimes to be able to start from a large value for the neighborhood width and progressively decrease it as time goes by. In such cases, the following formula was chosen for specifying σ at each iteration k:

$$\sigma(k) = \sigma_{\text{start}} \left(\frac{\sigma_{\text{end}}}{\sigma_{\text{start}}} \right)^{k/k_{\max}}. \tag{4}$$

2 Experiments

All the experiments in this report were performed on the MNIST dataset LeCun et al. (1998). It consists of 28×28 images of handwritten digits from 0 to 9. In fact, the dataset was restricted here to only the images corresponding to digits 5, 6, 7, and 9, according to a pseudo-random procedure provided in the Supplementary Materials.

The learning rate was set throughout as $\eta = 0.05$. Empirically, this value revealed itself to be small enough so that the Kohonen map reaches a local minimum of the fit function, and large enough so that it converges to this minimum in a reasonable amount of time. The maximum number of iterations was set to maxit $= 5000$, but as long as the average difference between prototype coordinates in successive iterations reaches below tol $= 10^{-5}$, the algorithm is considered to have converged.

Figure 1 shows the result of the fit given by Kohonen's algorithm for different grid sizes and neighborhood widths. For Figure 1(f), σ was decreased according to 1. Since the input datapoints are vectorized 28×28 images, the resulting prototypes can also be visualized as 28×28 images. The number that appear on top of each prototype image is the label of the input datapoint that most closely resembles this prototype. This was computed by taking the Euclidean distance of the prototype to each datapoint, and checking for which row of D this was minimized. The corresponding label is then found by looking into the same row, but in the label vector that accompanies the data matrix D.

All results can be reproduced by running the Jupyter Notebook provided in the Supplementary Materials. There might be small differences due to the fact that the prototypes are initialized in random positions at each run.

3 Discussion

We see in Figures 1(a) and 1(b) that if the neighborhood width is large in comparison to the grid size, then all prototypes will look very similar and tend towards the overall mean of the input data cloud. If we decide to fix σ throughout the run of the algorithm, its optimum value will depend on the grid size.

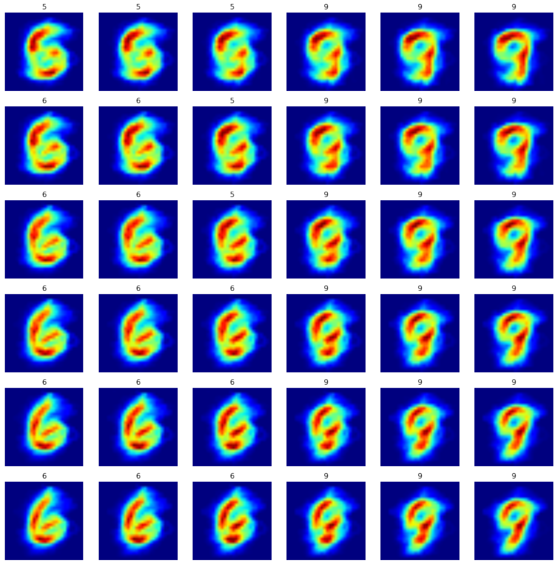
A good result is observed in Figure 1(d), where we see that different regions of the map fit to different digits, giving us a better representation of the data set.

However, one can argue that the best result lies in Figure 1(f), as every label is approximately proportionally represented on the prototype map. The decreasing procedure taken on σ is so that the initially randomly located prototypes quickly converge towards the center of the data cloud. Then, towards the final iterations, each prototype is less and less affected by its neighborhood, allowing itself to refine its position in the label cluster it is representing.

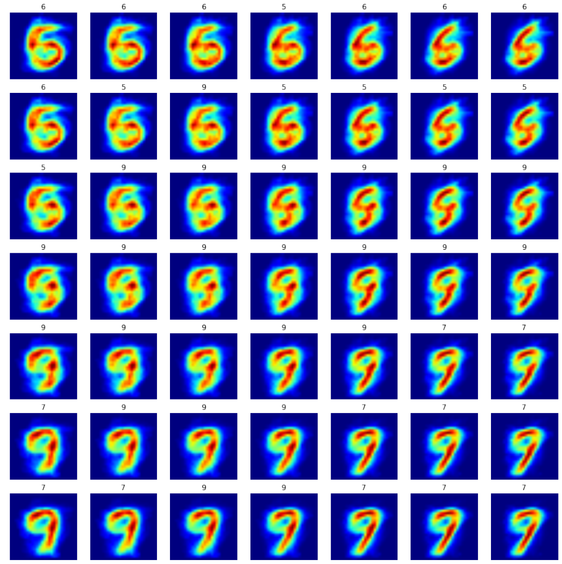
References

Kohonen, Teuvo. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982. ISSN 1432-0770. doi: 10.1007/BF00337288. URL `http://dx.doi.org/10.1007/BF00337288`.

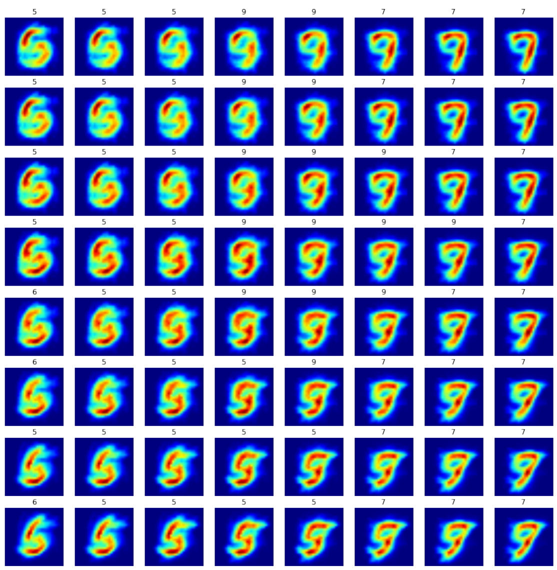
LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.



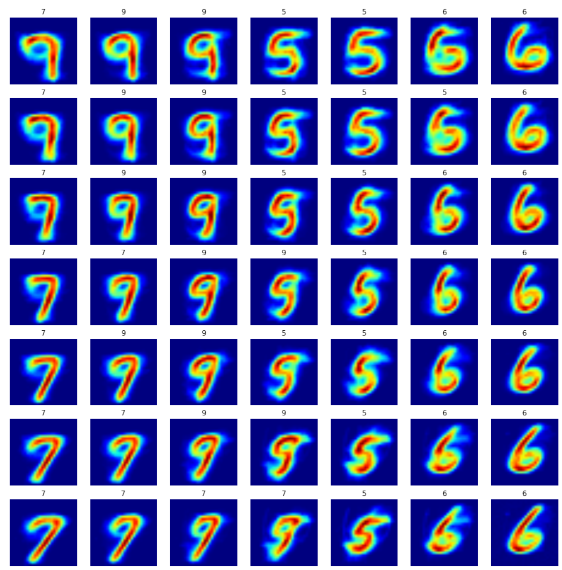
(a) 6×6 grid, $\sigma = 3$.



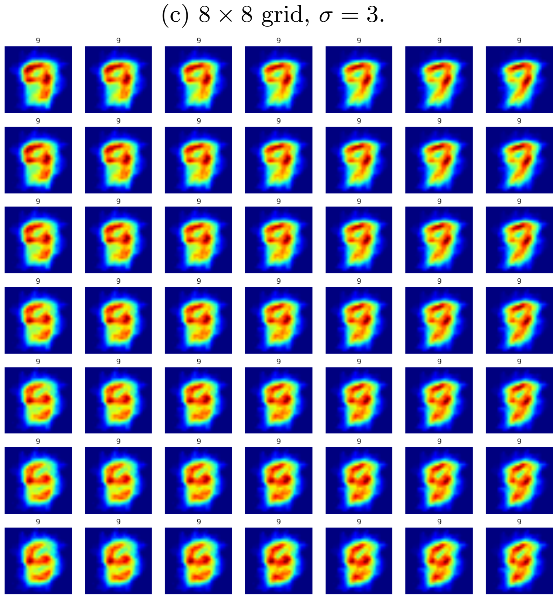
(b) 7×7 grid, $\sigma = 3$.



(c) 8×8 grid, $\sigma = 3$.



(d) 7×7 grid, $\sigma = 1$.



(e) 7×7 grid, $\sigma = 5$.

(f) 7×7 grid, σ decreasing from 5 to 0.1.

Figure 1: Prototypes found for different configurations of Kohonen's algorithm.