Introduction to Machine Learning
Assignment 4

Group 06
Lucas Pereira (s4507983) & Matteo Wohlrapp (s4974921)

October 11, 2021

Introduction

In this practical, we experiment with Vector Quantization, VQ. Vector Quantization is a form of unsupervised Winner-Takes-All competitive learning. The aim of the technique is to reduce data by representing the given data set S, which is usually comprised of vectors $x_1, .., x_n \in R^n, n \in N$, through fewer representatives. To achieve this, K prototype vectors $w^1, .., w^K \in R^N$ are initialized first. Afterwards, these prototypes are trained for t_{max} epochs. In each epoch, every vector $x_i \in S$ is considered and the closest prototype w^{i*} according to e.g. the squared euclidean distance $d[w, x] = \sum_{j=1}^{N}(w_j - x_j)^2$, is assigned to it: $w^{i*} = argmin_j\{d[w^j, x_i]\}$. Then, w^{i*} is moved according to the following formula: $w^{i*} = w^{i*} + \eta * (x_i - w^{i*})$, where η is considered as the learning rate and determines how much movement happens with one update step. The goal is to minimize a cost function given by $H_{VQ} = \sum_{j=1}^{K}\sum_{\mu=1}^{P}(x^\mu - w_j)^2\Pi_k^K\Theta(d_k^\mu - d_j^\mu)$, where

$$\Theta(x) = \begin{cases} 1, & \text{if } x >= 0. \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Methods

To implement the Vector Quantization, we use MATLAB. First we load the given data which contains one set of $1000x2$ double values, representing (x, y) coordinates in a $2D$ plane. The first step is to select the prototypes. Two methods to do so were implemented, one selecting random data points and one selecting the worst possible prototype positions. These are derived by calculating the mean of every vector $x_i \in S$ and then selecting the K vectors with the biggest distance to the position. Afterwards training is performed. In each epoch from 1 to t_{max} the data set S is permuted randomly and the following steps are executed for each $(x, y) \in S$:

1. Find the closest prototype: $w^{i*} = argmin_j\{d[w^j, x_i]\}$
2. Update w^{i*} according to $w^{i*} = w^{i*} + \eta * (x_i - w^{i*})$.

Furthermore, H_{VQ} is calculated in every epoch. Here, the following function is used:

$H_{VQ} = \sum_{j=1}^{K}\sum_{\mu=1}^{P}(x^\mu - w_j)^2\Pi_k^K\Theta(d_k^\mu - d_j^\mu)$, where Θ is defined as in (1). $\Theta(x)$ ensures that the prototype w_j is only considered for the error value when it is the closest to x^μ. The H_{VQ} for each epoch is collected and later plotted to show how the prototype training affects H_{VQ} over time. Furthermore we also keep track of the prototype position over time to plot the trajectory of the movement. As a bonus, we also analyze the the final H_{VQ} as a function of K, considering the values $K = 1, 2, 3, 4, 5$. To find the optimal t_{max} and η we wrote a function taking the learning rate and the number of prototypes as input parameters, then returning the last H_{VQ} value. This function is used to loop over η from 0 to 1, increasing with a step-size of 0.05 and over t_{max} from 0 to 2000, increasing with a step-size of 200. Afterwards, the results are plotted to analyze the optimum in an empirical manner.

Results

In this section we present the results of our program performing Vector Quantization. First we display the plots concerning the optimal η and t_{max}. Afterwards plots concerning the H_{VQ} for $K = 2$ and $K = 4$ are shown. Furthermore we present the trajectory for the prototype movement and the last H_{VQ} value of a function of K.

0.1. Varying parameters

In this section findings concerning the optimal η and t_{max} parameters are presented.

0.1.1. Varying η

Figure 1 in the appendix shows the final H_{VQ} value as a function of η. The graph decreases rapidly for $\eta \in [0, 0.05]$. Afterwards, η is constant for a while until the values starts to behave unstable after $\eta = 0.45$.

0.1.2. Varying t_{max}

Figure 2 in the appendix shows the final H_{VQ} value as a function of t_{max}. The graph has minimum at $t_{max} = 800$ but other than that shows no clear tendency but rather unpredictable behaviour for different values of t_{max}.

0.2. H_{VQ} for $K = 2$ and $K = 4$

In this section, H_{VQ} is plotted as a function of t. Each plot uses $t_{max} = 800$, which we derived as the optimal value from varying the parameters, and shows the error for $K = 2$ and $K = 4$. η is varied in all three plots, with $\eta \in \{0.01, 0.1, 0.9\}$. These values were selected after evaluating how η influences H_{VQ} in the previous section.

Figure 3 in the appendix shows H_{VQ} as a function of t with $\eta = 0.01$. At this configuration, the difference between K_2 and $K = 4$ is significant. The function for $K = 2$ is relatively stable, while for $K = 4$, it decreases fast after around 180 epochs.

Figure 4 in the appendix shows H_{VQ} as a function of t with $\eta = 0.1$. At this configuration, the difference between K_2 and $K = 4$ is also significant. The function for both K is relatively stable in terms of average, but has big spikes in between.

Figure 5 in the appendix shows H_{VQ} as a function of t with $\eta = 0.9$. At this configuration, the difference between K_2 and $K = 4$ is small compared to the spikes in the function, however, $K = 4$ still generates a better result.

0.3. Prototype trajectory

Figure 6 and 7 in the appendix shows the original points in for each prototype as a green square, the final points as a pink square and the prototype values in between as smaller blue X. As we can observe in both pictures the prototypes, no matter how close they are to themselves at the start, seem to try to be equally distant to each other while keeping themselves inside the closest cluster of points. This is due to the way we move our prototypes. By moving only the winner, we give them a "bias" to stay in a selected area. This happens even in the worst case initialization, where three points try to equidistantly nest themselves in the closest cluster of points while one is "forced out" by the further away points. We can also see, that the movement towards the second cluster by a single prototype is quite fast because of the sheer number of them "attracting" it towards them, since he most likely "wins" for all points in that second cluster. Its fast "closeness", ensures that no other point will make the travel, since it will always be the closest point to those points.

0.4. Final H_{VQ} as a function of K

As a bonus, we included a function of the final H_{VQ} value as a function of K. Figure 8 in the appendix shows the function for $K = 1, 2, 3, 4, 5$. We used $\eta = 0.1$ and $t_{max} = 800$ as derived in the previous section as the optimal values. The graph decreases fast from $K = 1$ to $K = 2$, but the decline slows afterwards. Using the 'elbow method', $K = 2$ would be the optimal amount of prototypes.

Discussion

When considering the optimal value for the learning rate, the results are as expected. Values which are too small or too big give inaccurate results. However, the reason for this behaviour differs for both configurations. When η is too small, the prototypes update too slow for the used t_{max}. This also explains, why the descent in Figure 1 is linear until roughly $\eta = 0.05$ is reached. The training was simply not extensive enough to get sufficient accuracy. In comparison, when η is too big, the distance at each update step creates inaccuracy since the prototype is moved further than the optimal position. However, this optimal position depends heavily on the data set and therefore the final H_{VQ} behaves chaotic for increasing η. t_{max} on the other hand has a far more unpredictable input on the result. We expected a higher accuracy with increasing t_{max}, however it appears the optimal value for the data set with η of 0.1 is 800. This can be due to the fact that with increasing t_{max}, the prototype positions move in a way which does not complement the minimization of H_{VQ}. When regarding different values for K, mainly $K = 2, 4$, the results were as expected. The higher the amount of prototypes, the lower the overall H_{VQ}. This intuitive assumption, is backed by our experiments, Figure 3 to 5, where the general trend for the H_{VQ} is lower for $K = 2$ than for $K = 4$. The before mentioned unstable behaviour for larger η is also observable in these 3 figures. The larger η, the greater fluctuations in H_{VQ}. Comparing Figure 5 to Figure 3, the spikes in the graph increase drastically for $\eta = 0.9$ than for $\eta = 0.01$.

We expected a clear decreasing trend for H_{VQ} with increasing t, but the graphs indicate otherwise. The only clear sign of decreasing H_{VQ} can be found in Figure 3 with four prototypes. This behaviour can be explained by the trajectories of the prototypes, as they tend to try and remains equidistant, having the most points for "themselves", the more prototypes you add the smallest the area for each prototype will probably be (the random selection of points may cause some prototypes to have huge areas, see figures 6 and 7). Since H_{VQ} is measured by adding the distance towards winning prototypes, the more prototypes following this "equidistant behavior", the smallest this value will be.

The before mentioned increase in accuracy with increasing K is also shown by Figure 7. For each increase in K, the final H_{VQ} value decreases. The function is not linear, with the biggest decrease happening from $K = 1$ to $K = 2$. Using the elbow technique, $K = 2$ would be the optimal pick. Overall, the graphs were as expected, with higher K and lower η yielding higher accuracy.

Workload

The workload was split equally, with Lucas Pereira and Matteo Wohlrapp working both on code and report design. Issues and problems were solved in a collaborative manner, with both team partners contributing evenly.

A. Assignment 4

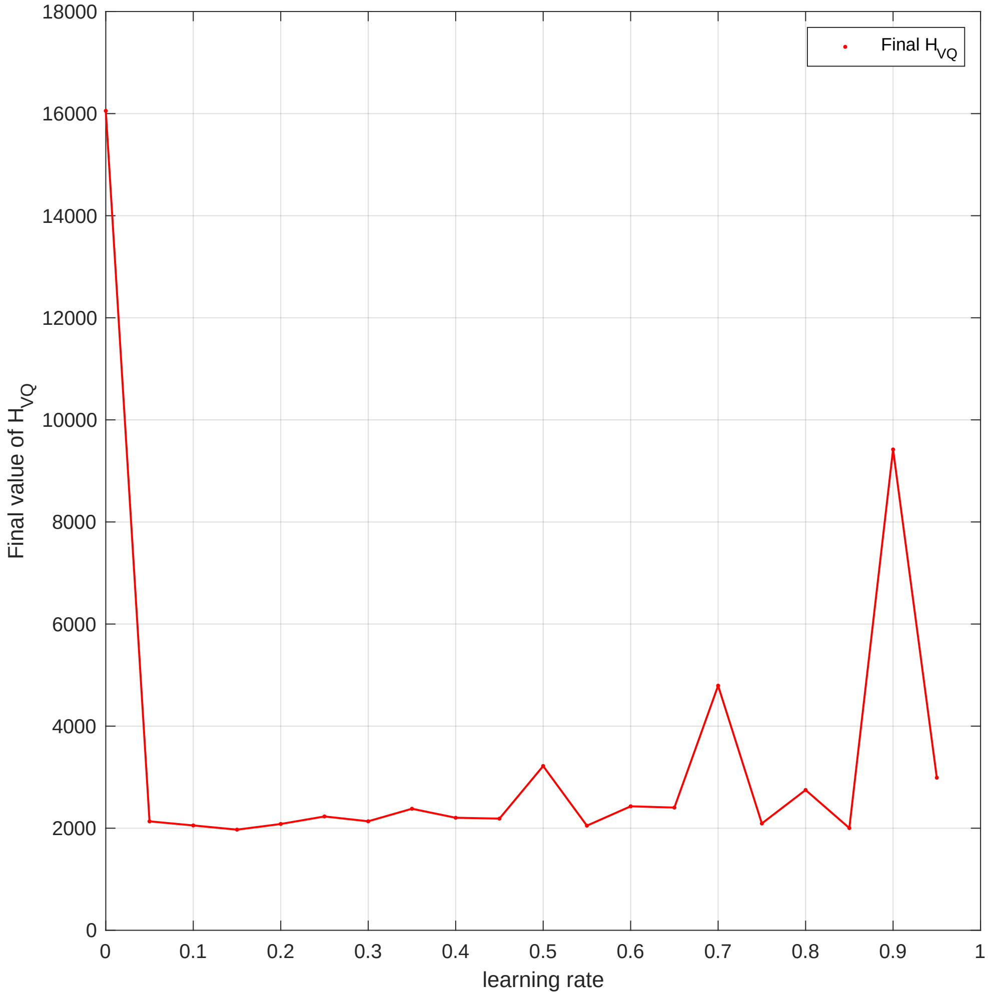

Figure 1: *Graph of the final H_{VQ} as a function of η*

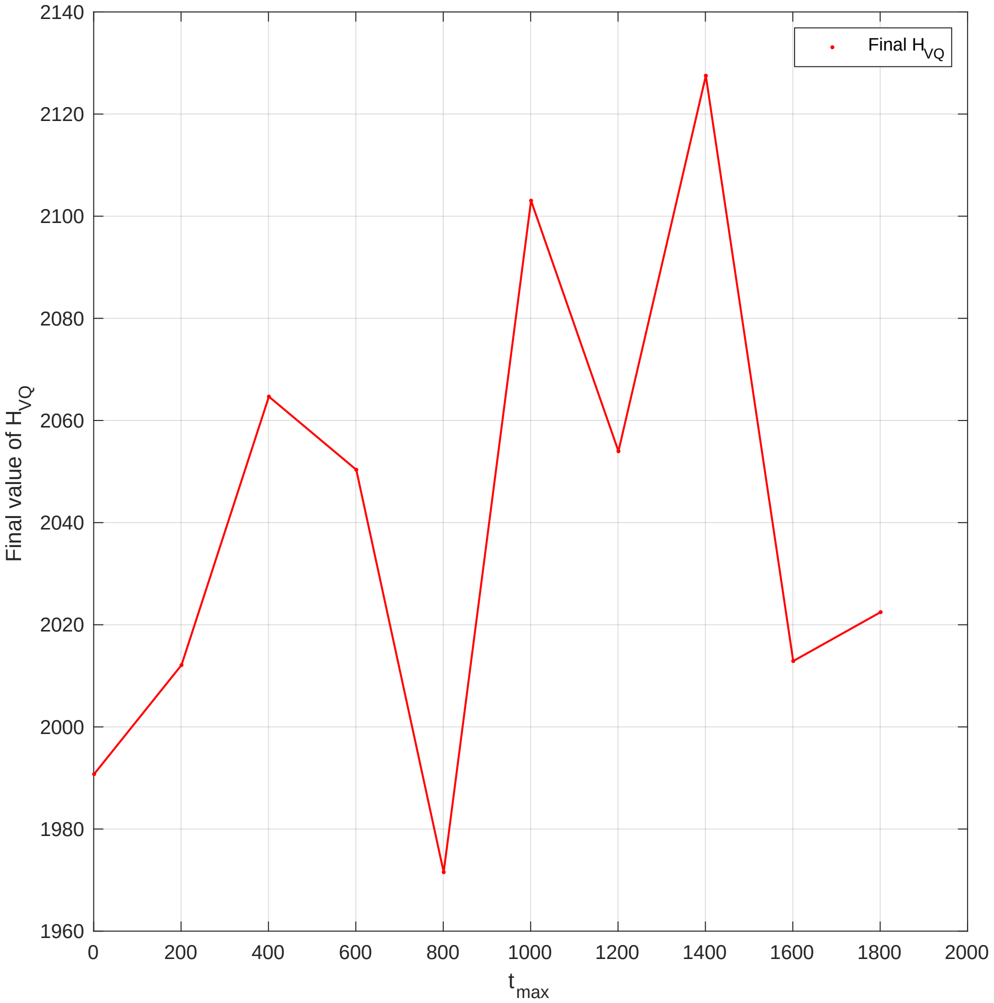

Figure 2: *Graph of the final H_{VQ} as a function of t_{max}*

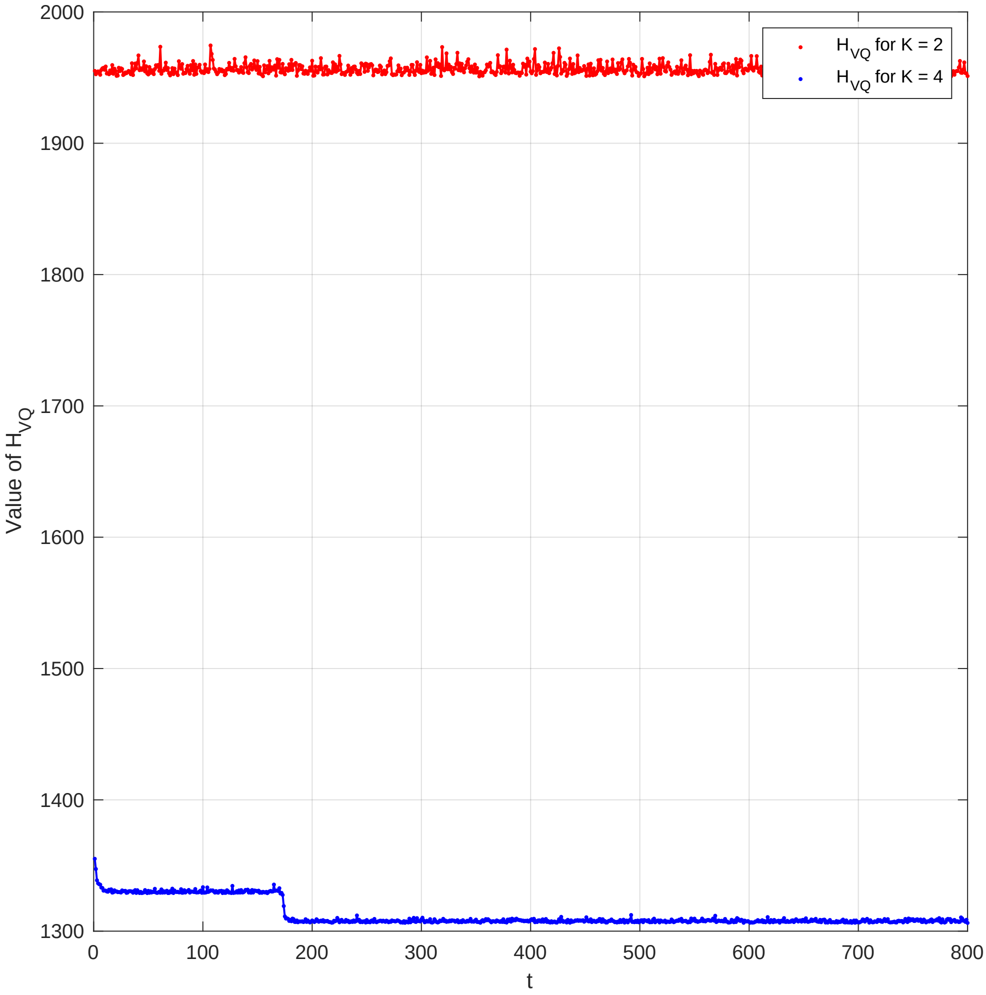

Figure 3: *Graph of* H_{VQ} *and* t *with* $\eta = 0.01$

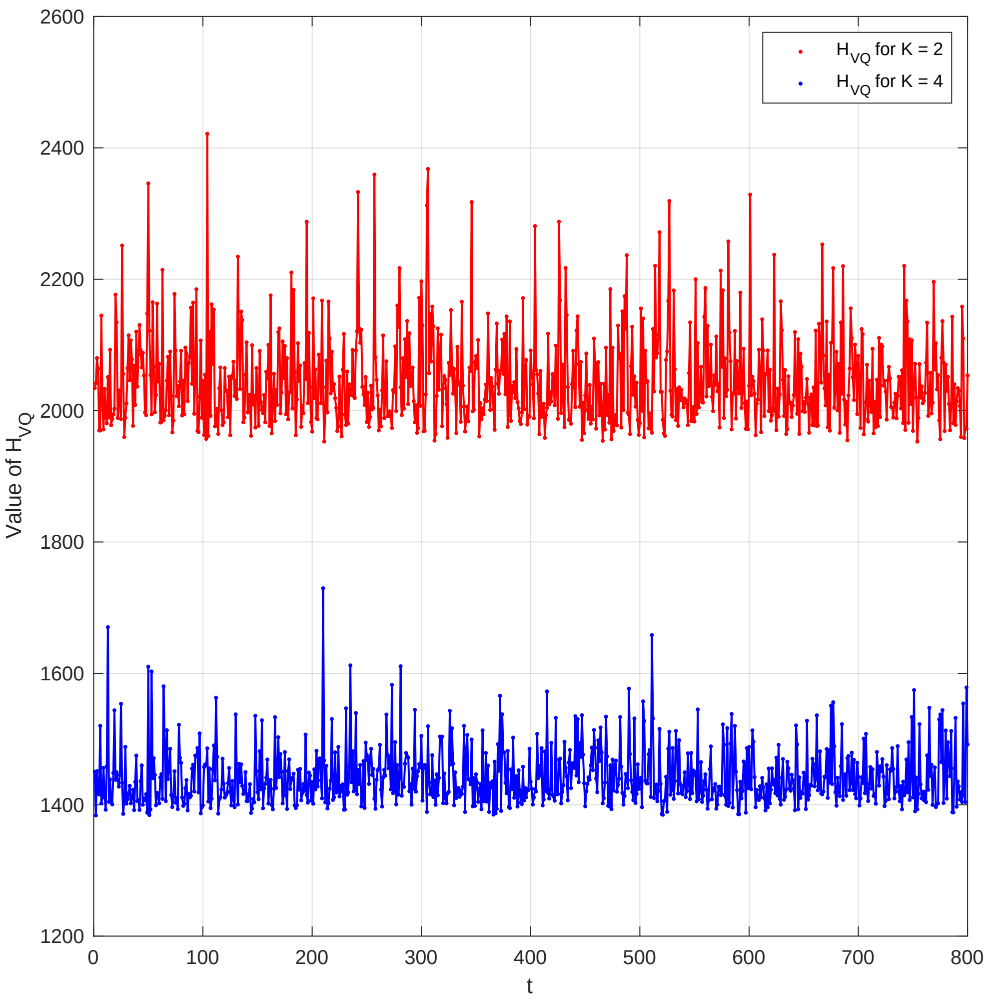

Figure 4: *Graph of H_{VQ} and t with $\eta = 0.1$*

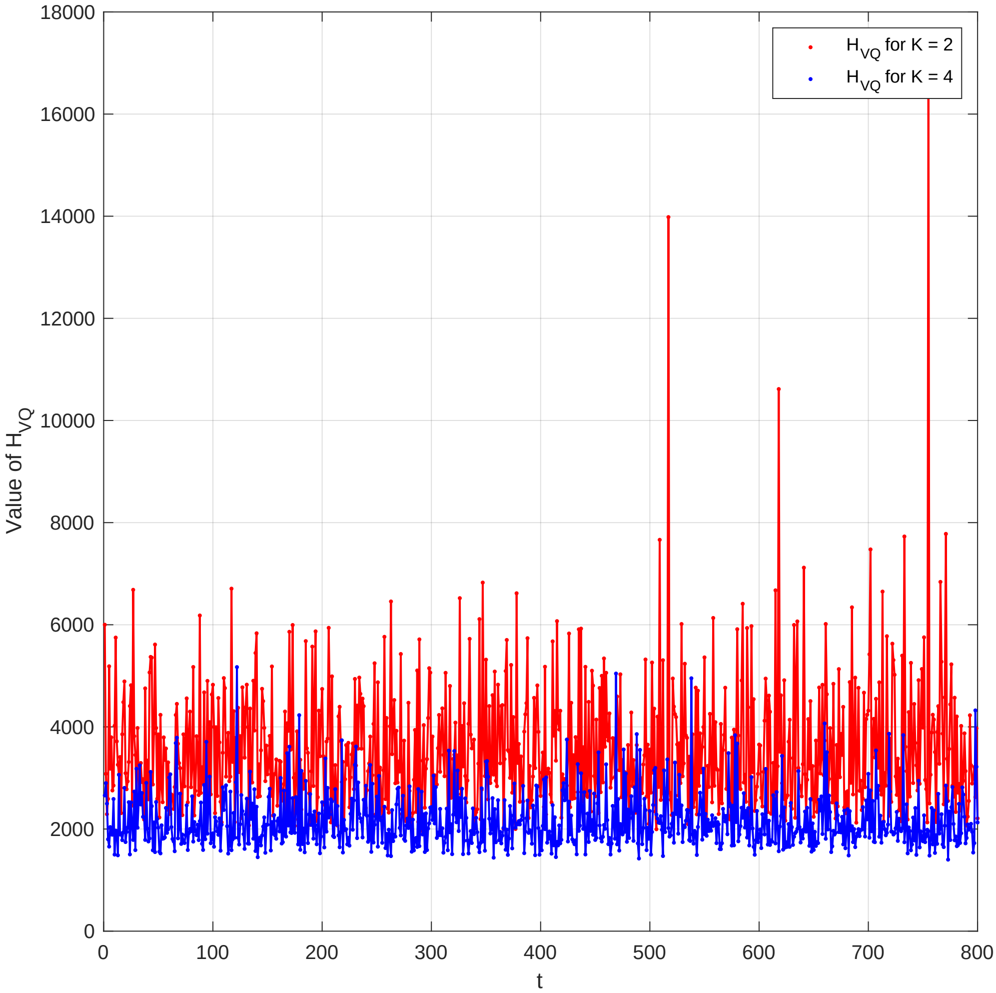

Figure 5: *Graph of H_{VQ} and t with $\eta = 0.9$*

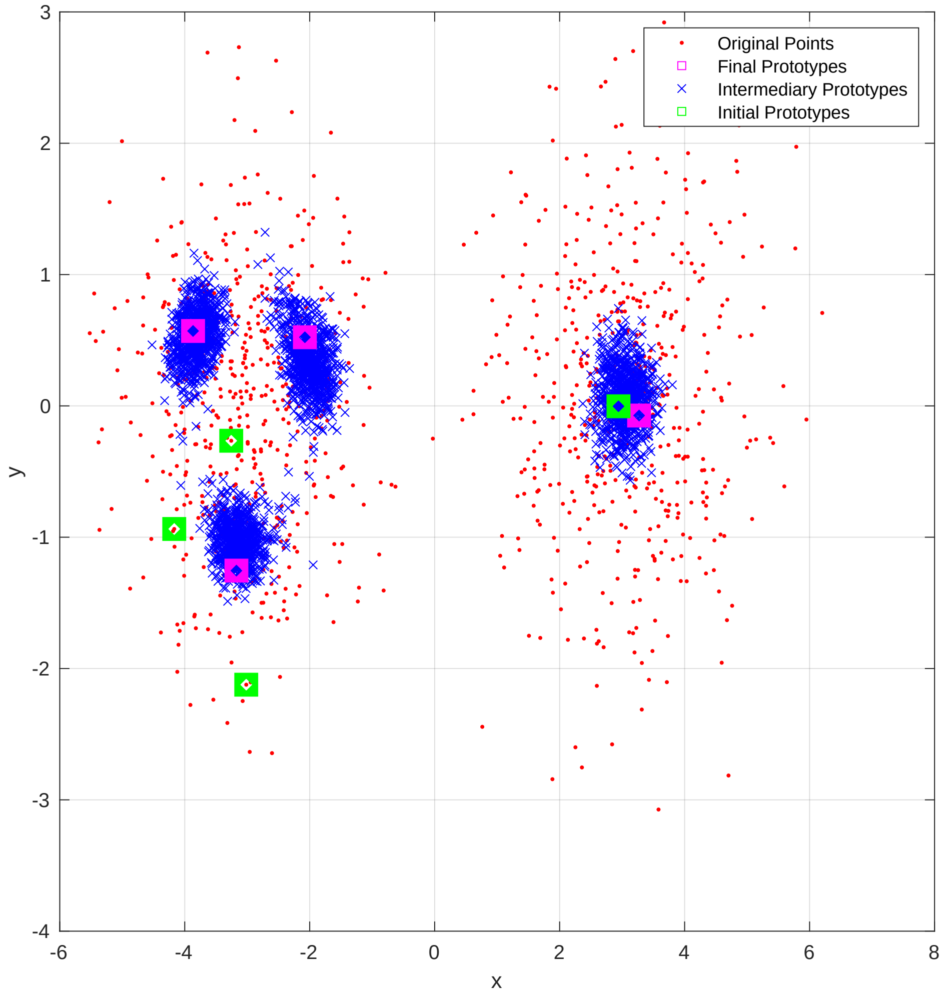

Figure 6: *Graph of the trajectories of randomly selected prototypes with* $K = 4$

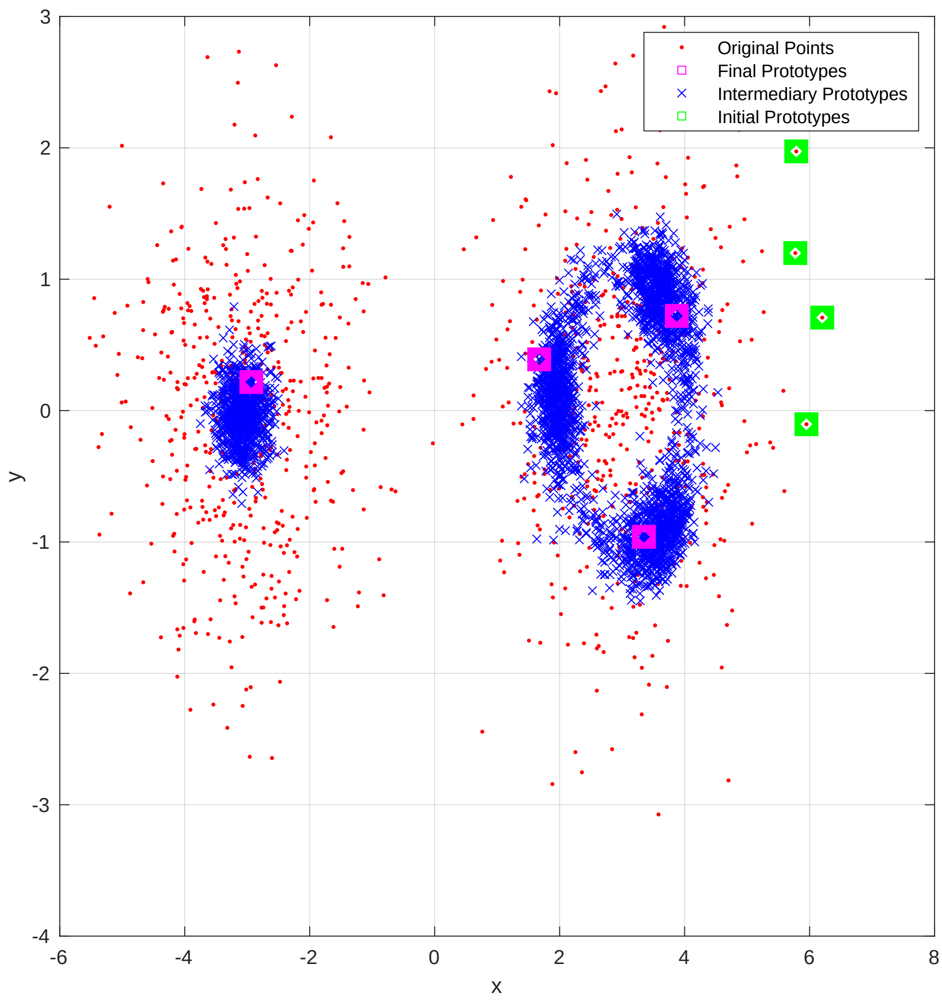

Figure 7: *Graph of the trajectories of bad selected prototypes with* $K = 4$

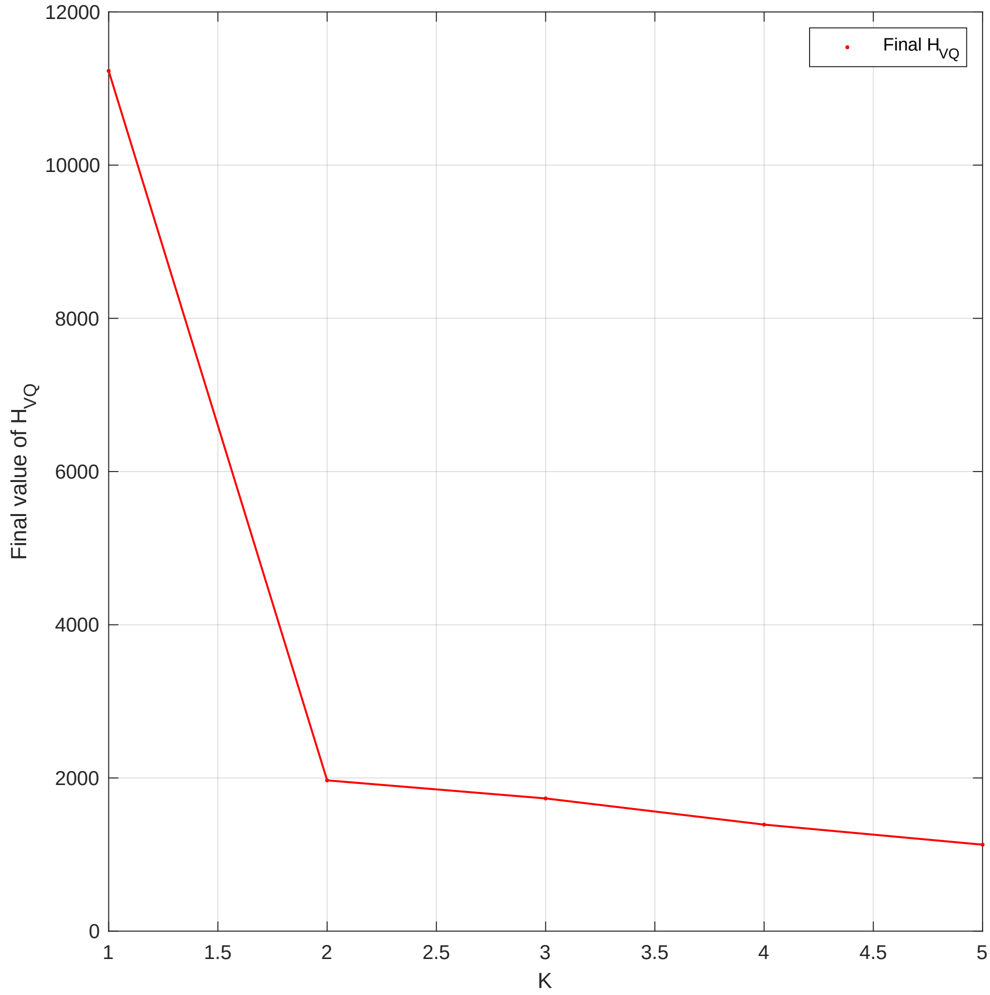

Figure 8: *Graph of the final H_{VQ} as a function of K with $\eta = 0.1$ and $t_{max} = 800$*