Introduction to Machine Learning
Assignment 4

Group 31
Stijn Kammer (s4986296) & Ramon Kits (s5440769)

October 13, 2022

1. INTRODUCTION

The method used in this assignment is Winner-Takes-All unsupervised competitive learning Vector Quantization. The use of this method is to classify the data into clusters and to find the center of each cluster. These centers are used to classify the data into the clusters making it possible to classify new data. The Winner-Takes-All refers that one prototype is chosen to represent the cluster and the other prototypes are pushed away from the cluster into the other clusters. In this assignment we use the simplevqdata dataset to test the method. The dataset contains 1000 data points with 2 features.

2. METHODS

Vector Quantization makes use of prototype vectors to represent the data and is often used for identification and grouping of clusters in similar data. To use Vector Quantization, we first need to define a set of prototype vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$. The prototype vectors are chosen by selecting k data points from the data set. After which present a single example $\mathbf{x}$, we can find the closest prototype vector $\mathbf{v}_i$ by minimizing the distance between $\mathbf{x}$ and $\mathbf{v}_i$. The distance between $\mathbf{x}$ and $\mathbf{v}_i$ is defined as the Euclidean distance between the two vectors. When the closest prototype vector $\mathbf{v}_i$ is found, we move the prototype vector $\mathbf{v}_i$ towards $\mathbf{x}$ by a fraction η of the distance between $\mathbf{x}$ and $\mathbf{v}_i$. Repeat this process for all the examples in the training set and the prototype vectors will converge to a set of vectors that represent the data well. Every time all datapoints are used to update the prototype vectors, this is called an epoch. The prototype vectors are then used to classify new examples.

Quantization error is a number that is calculated when using the prototype vectors to classify new examples. To be able to estimate a good value for k, we can use the quantization error to find the optimal value for k, the error should be as low as possible while also being stable. Stability is important because it means that the error is not suddenly going up a lot when data is added or removed. If the error is not stable, it means that whenever the data changes, a new optimal number of epochs is needed to be found. The quantization error is defined as the sum of distances between the prototype vectors and the examples in the training set. The formula for the quantization error is given by:

$$H_{VQ} = \sum_{j=1}^{K} \sum_{\mu=1}^{P} (x_\mu - w_j)^2$$

where K is the number of prototypes, P is the number of examples in the training set and w_j is the prototype vector.

The quantization error is minimized by moving the prototype vectors towards the examples in the training set. The quantization error is also minimized by choosing a good set of prototype vectors. The prototype vectors should be chosen in a way that they are spread out over the data set. This is done by choosing the prototype vectors to be the k data points that are furthest apart from each other. The prototype vectors should also be chosen in a way that they are representative of the data set. This is done by choosing the prototype vectors to be the k data points that are closest to the mean of the data set.

3. Learning curves

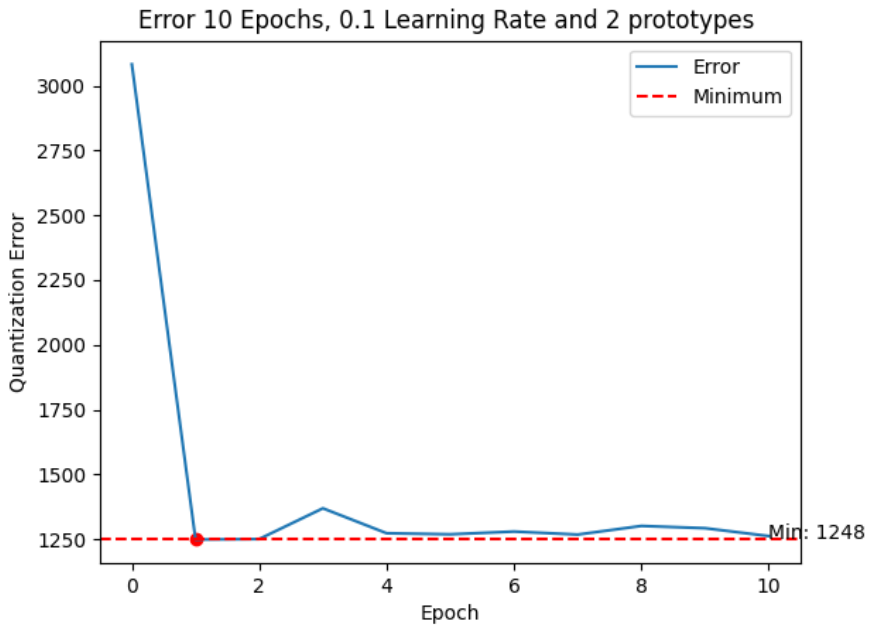

Figure 1: *Error for $t_{max} = 10$, $\eta = 0.1$ and $K = 2$*

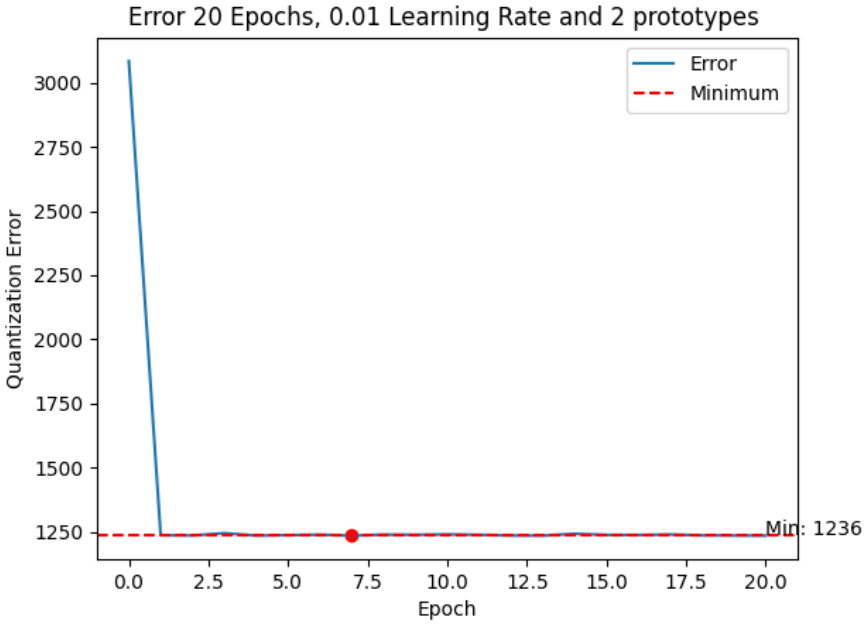

Figure 2: *Error for* $t_{max} = 20$, $\eta = 0.01$ *and* $K = 2$

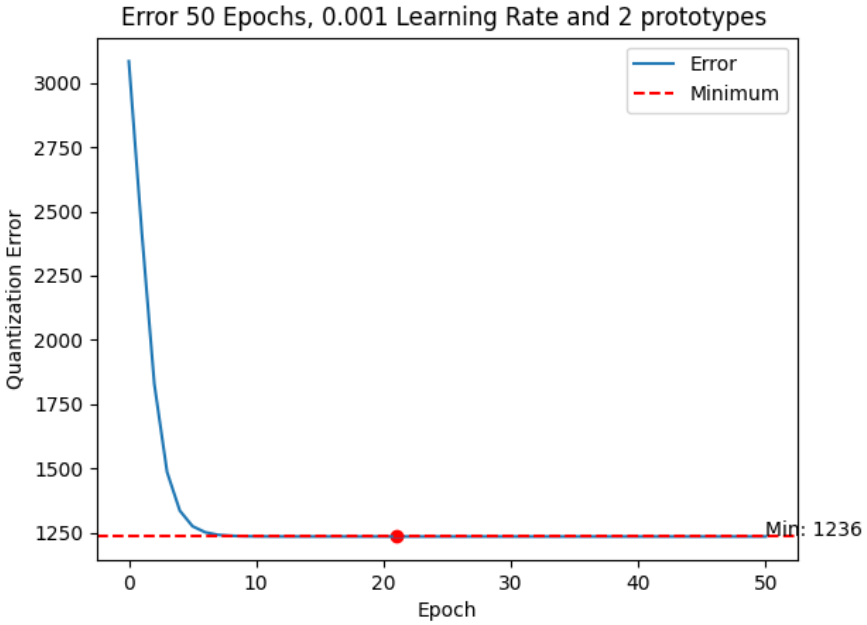

Figure 3: *Error for* $t_{max} = 50$, $\eta = 0.001$ *and* $K = 2$

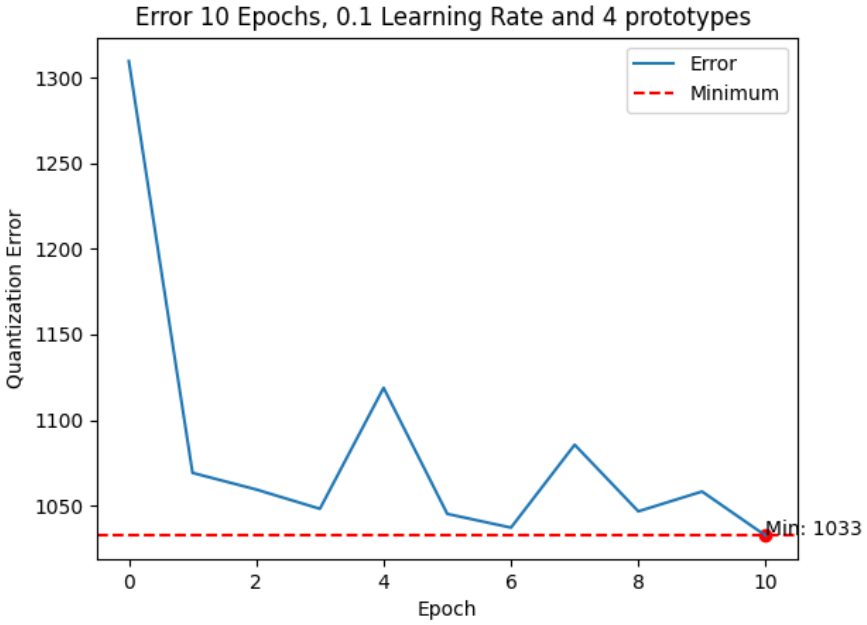

Figure 4: *Error for* $t_{max} = 10$, $\eta = 0.1$ *and* $K = 4$

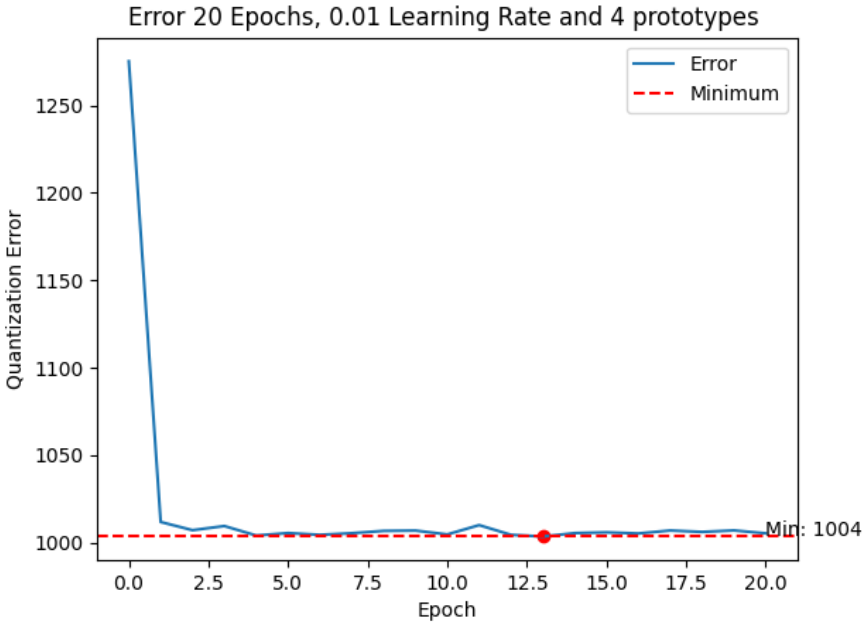

Figure 5: *Error for* $t_{max} = 20$, $\eta = 0.01$ *and* $K = 4$

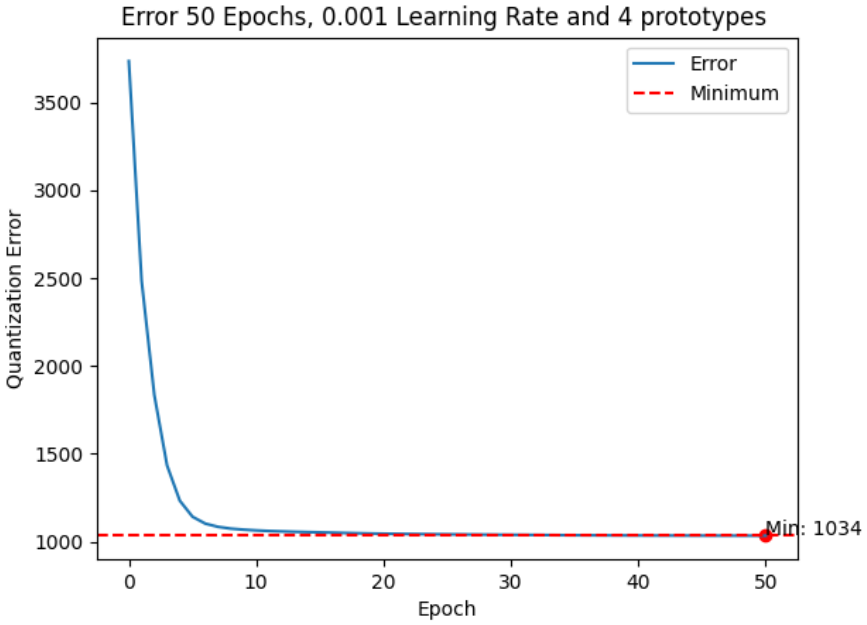

Figure 6: *Error for* $t_{max} = 50$, $\eta = 0.001$ *and* $K = 4$

4. Trajectories of prototypes

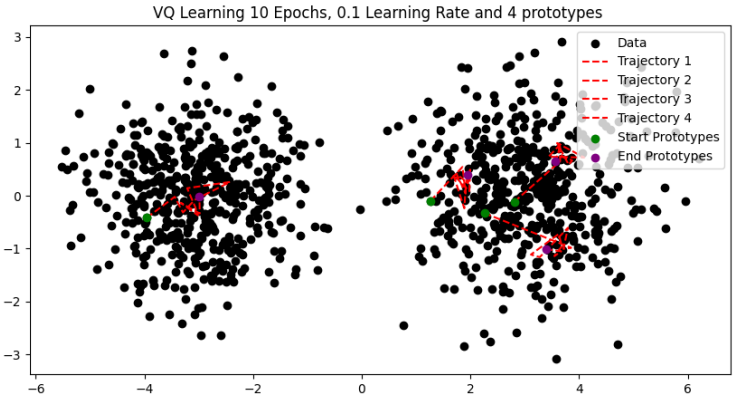

Figure 7: *Trajectories of prototypes for Vector Quantization with* $t_{max} = 10$, $\eta = 0.1$ *and* $K = 4$

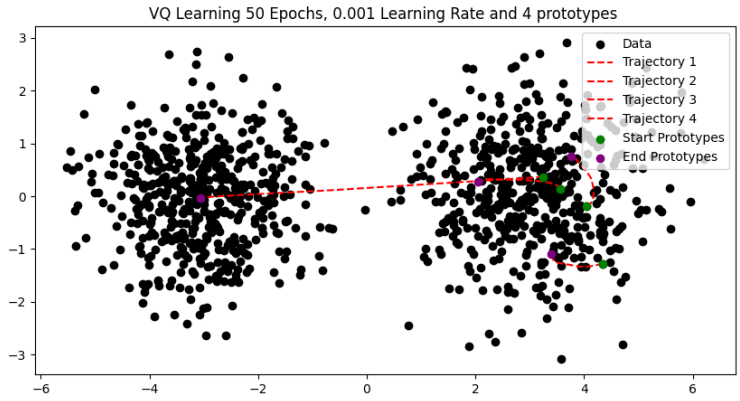

Figure 8: *Trajectories of prototypes for Vector Quantization with* $t_{max} = 50$, $\eta = 0.001$ *and* $K = 4$

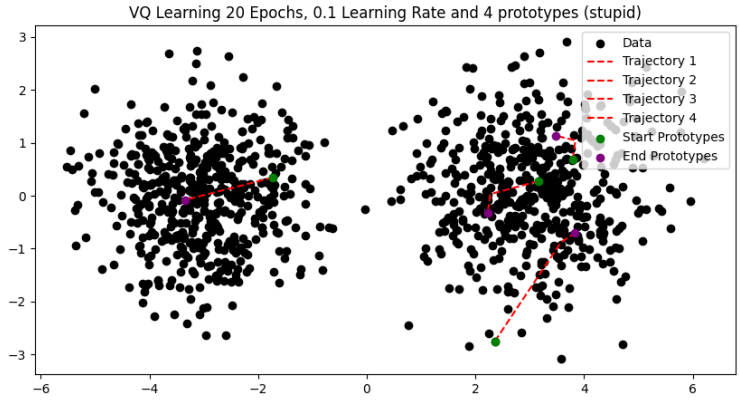

Figure 9: *Trajectories of prototypes for Vector Quantization with* $t_{max} = 20$, $\eta = 0.1$ *and* $K = 4$

5. Discussion

Learning curves

For $K = 2$ in Figure 1 and Figure 2 we can see th error is decreasing very fast. Already after 1 epoch the error seems to be very low and after that it is not hardly getting any lower. This is because due to the high learning rate η the prototypes are moving very fast towards the data points. When it is needed to do very view epochs, this can be a good thing, but this is almost never the case. The error is also not stable, which means that the optimal number of epochs is not the same when the data changes. the point at 7 epochs may be optimal in this example with exacly this data set, but it might not be optimal for an altered version.

In Figure 3 we can see that the error is decreasing very slowly. This is because the learning rate is very low. This means that the prototypes are moving slowly towards the data points. Because of this, it is needed to do many epochs to get a good result. But when a good result is achieved, it is very stable. When the data changes, the error will not change a significant amount. All three reach around the same error, but using $\eta = 0.001$ gives the most relieable results.

Where with $K = 2$ we saw a relatively stable error with a high η, with $K = 4$ in Figure 4 and Figure 5 we see a more volatile error with a high η. This is because since there are more prototypes, they are more likely to be close to each other and therefore are closer to some datapoints in one epoch but are not in the other, which make them make big moves at semi-random moments. This makes analizing the dataset with a high η even less usefull.

In Figure 6 we see that the error is decreasing very slowly and being stable when it nears its minimum. This illustrates that with a low η, the chances of having a good working model are higher.

Trajectories of prototypes

Figure 7 and Figure 8 perfectly demonstrate the effect of having a high η versus a low η. The high η makes the prototypes move very fast towards the data points, which makes the error decrease very fast. But it also makes the prototypes change position very fast once it approaches a good position. This makes the quantization error unstable and the model less reliable. The low η makes the prototypes move slowly towards the data points, which makes the error decrease slowly and stay stable once it reaches a good position.

Also the vector quantization with $K = 4$ and a "stupid" approach has been tested, This means that the datapoints are not shuffled every epoch. The order in which the datapoints are presented to the model is the same every epoch. As to be seen in Figure 9 the prototypes are moving fast towards one point and after that they are hardly moving at all. This is because the datapoints are presented in the same order every epoch, which means that the prototypes are moving towards the same datapoint every epoch. After not many epochs they reach a stop point at which they are not moving anymore because every epoch is a copy of the previous one, which means that the prototypes are not moving anymore because they are already at the optimal position. This is not beneficial for the model, because it will not try other positions and therefore will not find the very optimal position.

6. Work distribution

This week the work has been divided relatively between the group members as follows: Ramon has written the majority of the code and Stijn wrote the methods section after together examining the code and functionality of it. All other parts have been written together. The final report is scanned by both Ramon and Stijn and they have discussed and implemented the possible changes. Both group members have learned how to apply the vector quantization algorithm on a dataset and how to analyze it.