\documentclass[twoside, a4paper, fleqn, reqno]{article}

\usepackage[

labNumber=3,

authorZero={Lucas Pereira},

studentNumberZero={4507983},

authorOne={Matteo Wohlrapp},

studentNumberOne={4974921},

groupNumber={06}

]{intelligentsystems}

\begin{document}

\maketitle

\section\*{Introduction}

In this practical, we experiment with validation procedures for Learning Vector Quantization, $LVQ1$. Learning Vector Quantization is a form of supervised learning which uses n-dimensional data vectors split into multiple classes to derive prototypes from the data set and then use distance based classification such as the squared Euclidean distance to train the prototypes. In the training process the prototype positions are modified depending on the correct or incorrect association with the classes of neighboring data points.

To validate this process, we use an approach which splits the training set in $P$ disjoint sets: $D = \bigcup\limits\_{i=1}^{P} D^{(i)}$. Then we derive two separate sets: the training set $D^{(i)}\_{train} = D/D^{(i)}$ and the testing set $D^{(i)}\_{test} = D^{(i)}$. We use these to train the prototypes $P$ times for every possible variation of testing and training data and calculate the resulting errors for both data sets.

\section\*{Method}

To implement the validation we use MATLAB. First we load the given data which contains one set of $100 x 2$ values, representing $(x,y)$ coordinates in a $2-D$ plane.

The first step \*removed\* is to classify the data as shown in previous labs. Afterwards we split the data set according to the number of $P$ values. Splitting $D$ into $P$ sets leaves us with $P$ different possibilities to retrieve $D^{(i)}\_{train}$ and $D^{(i)}\_{test}$. Therefore we created a loop iterating over $i \in \{1,..,P\}$, every time selecting a different set for $D^{(i)}\_{test}$.

We then choose $K$ prototypes from $D^{(i)}\_{train}$ and train them for 100 epochs. In each epoch, we regard every data point and search for the closest prototype with respect to the squared Euclidean distance: $w^\* = argmin\_j\{d[w^j, x^m]\}$. Then we update the position of $w^\*$ according to the following formula: $w^\* = w^\* + \eta\_w \* \Psi(S^\*, \sigma^m) \* (x^m - w^\*)$. Here, $\eta\_w$ is considered the learning rate and the function $\Psi(S^\*, \sigma^m)$ is defined as

\begin{equation}

\Psi(S^\*, \sigma^m) =\begin{cases}

1, & \text{if $S=\sigma$}.\\

-1, & \text{otherwise}.

\end{cases}

\end{equation}

After the training, we use the derived prototypes and $D^{(i)}\_{test}$ to validate the results and plot the error rate.

For this particular lab, we first consider a partition with $P = 5$, meaning $|D^{(i)}\_{train}| = 80$ and $|D^{(i)}\_{test}| = 20$. As a bonus we additionally implemented cross-validation with $P = 10$, resulting in $|D^{(i)}\_{train}| = 90$ and $|D^{(i)}\_{test}| = 10$.

\section\*{Results}

In this section we present the results of our program performing cross-validation for $LVQ1$. We repeat each experiment $P$ times, with the bar at each calculated value showing the highest and lowest error value for the respective $K$ and the line representing the average error \*removed\*.

\subsection{$P = 5$}

The following figures show $D^{(i)}\_{train}$ and $D^{(i)}\_{test}$ with $P = 5$ for different numbers of prototypes $K$ per class.

\begin{figure}[H]

\centering

\includegraphics[width=\linewidth]{Train\_errors\_P5.pdf}

\caption[Caption for LOF]{Graph of $D^{(i)}\_{train}$ and $K$ with $P = 5$}

\end{figure}

\clearpage

\begin{figure}[H]

\centering

\includegraphics[width=\linewidth]{Test\_errors\_P5.pdf}

\caption[Caption for LOF]{Graph of $D^{(i)}\_{test}$ and $K$ with $P = 5$}

\end{figure}

\clearpage

\subsection{$P = 10$}

As a bonus, we also calculate and plot the errors for $P = 10$. The following figures show $D^{(i)}\_{train}$ and $D^{(i)}\_{test}$ for different numbers of prototypes $K$ per class.

\begin{figure}[H]

\centering

\includegraphics[width=\linewidth]{Train\_errors\_P10.pdf}

\caption[Caption for LOF]{Graph of $D^{(i)}\_{train}$ and $K$ with $P = 10$}

\end{figure}

\clearpage

\begin{figure}[H]

\centering

\includegraphics[width=\linewidth]{Test\_errors\_P10.pdf}

\caption[Caption for LOF]{Graph of $D^{(i)}\_{test}$ and $K$ with $P = 10$}

\end{figure}

\clearpage

\section\*{Discussion}

As expected, is the variation for the error calculated for $D^{(i)}\_{test}$ higher than the error for $D^{(i)}\_{train}$. This is as result of the smaller number of testing data, compared to training data. Before we perform the training, we shuffle the data set randomly, which can lead to an inhomogeneous distribution of falsely classified data at verification. Since $|D^{(i)}\_{test}| < D^{(i)}\_{train}$, this unequal distribution is more likely in $D^{(i)}\_{test}$.

Also, the huge differences between the lowest and highest error for $D^{(i)}\_{test}$ is as anticipated. When regarding the error values for Figure 2 or 4 at $K=2$, the margin between the highest and lowest value is quite high in comparison to the differences in Figure 1 and 3.

Also, we anticipated a constant error ration for different $K$. That is because the error rate is mostly connected to the number of epochs and not the number of prototypes as shown in Figure 5.

\begin{figure}[H]

\centering

\includegraphics[width=\linewidth]{Different\_prototypes.pdf}

\caption[Caption for LOF]{Graph of the error rate with different numbers of prototypes}

\end{figure}

\clearpage

At 100 epochs, the error rate is mostly stable, approaching the same value for every tested number of prototypes per class.

\section\*{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.

\printbibliography

\clearpage

%\appendix

%\input{appendix.tex}

\end{document}