Project 1: Won't you be my (K-Nearest) Neighbor

MTH448

Due: 2/18/2025

1 Project Description

In this project you will use the KNN (K Nearest Neighbors) algorithm described in class in order to classify hand-written digits.

1.1 MNIST Database

The MNIST database i contains 60,000 images of hand-written digits (converted into 28×28 pixel images) and labels of what the digit was supposed to be (see the notes from Lecture 4 for a discussion of how the database is formatted). Some sample entries in the database are presented below:

1.1.1 Project Objectives

1.1.2 KNN algorithm

Implement the KNN algorithm in a function *KNN()*. The function should receive the following parameters:

- *training_data*: a two-dimensional numpy array that contains the training data (each row is an element of the training data).
- *training_labels*: a one dimensional numpy array with the labels corresponding to the training data (the n^{th} element of *training_labels* is the label of the data in the n^{th} row in *training_labels*).
- *point*: a one dimensional numpy array containing the values of the point you wish to classify.
- *k*: the number of nearest neighbors that are to be used in the classification.

The result of your function should be the tuple (*lab*,*neibs*) where, *lab* is the predicted label of your given point and *neibs* is a list of row numbers of *training_data* that represent the *k* nearest neighbors of *point*.

Note: The KNN algorithm MUST be implemented from scratch, DO NOT use modules where the algorithm has already been implemented.

1.1.3 Algorithm Exploration

Explore the behavior of the KNN algorithm using the MNIST data and describe your results.

Some possible questions you may wish to consider are:

- How does the accuracy of the results depend on the size of the trining data set.
- How does the accuracy of the results depend on K (the number of nearest neighbors used in classification)?
- What happens if you use a metric (definition of distance) other than the Euclidean distance?
- Consider a weighted-KNN algorithm: instead of looking at the most common label, give each neighbor's label a weight based on its distance from the point (closer neighbors correspond to higher weights) and use the weights to select a label.
- Analyze examples of cases where the classification was wrong.
- What fraction of images were classified with perfect certainty (the labels of all of their neighbors were the same)
- Were there any cases where none of a given images neighbors had the correct label?
- Anything else that interests you.....

2 Grading

The report grade will have the following distribution:

Element	Weight	What will be graded
Introduction Section	10%	Quality of narrative.
Conclusion Section	10%	Quality of narrative.
Report Content	30%	Work done on developing the project. Your analysis, insights, observations, and interpretations. Quality of the narrative of the report.
Python Code	30%	Quality of the Python code included in the report. Relevance of the code to the project. Code organization and readability. Documentation of code by code comments.
Presentation	20%	Organization of the report. Text formatting. Use of LaTeX to typeset mathematics. Formatting of code output. Quality of graphs and plots.

2.1 Introduction

The introduction is the first section of your report. It describes the project (the underlying math) and the goals of the report. It should be written in a way that is engaging and understandable to a student who has some background in math and coding but does not take this course. The introduction SHOULD NOT be in list form. While some of the concepts in the introduction will be repeated from the project description, you should state the concepts in your own words and not copy from the project description.

2.2 Conclusion

This section should summarize your results and major findings. It can also include potential future extensions of the project. This should be the last section of the report.

2.3 Content and presentation

Outside of the two section mentioned your report should be broken up into sections and subsections in such a way as to maximize readability and comprehensibility. Where appropriate use Latex to format math.

Please note, projects ARE NOT just sets of coding exercises. They are reports in which you explore a particular math problem or phenomenon. Projects should be readable and interesting for a person not taking this class (and thus not having access to the project description) but familiar with math and python.

2.4 Code

Your code should be readable and understandable. Code should be broken up into small snippets each of which gets its own cell. Words should be used to describe what the code is doing and what the logic of your approach is. DO NOT put all of your code into a single section with no words or discussion.

Code should be readable with understandable variable names and comments included to explain what is not clear.

All python code included in the report should be written in such a way that it can be executed sequentially. For example, a function should never be used prior to its definition.

All code included must work. Do not submit code with errors.

Code output must serve the narrative of your report and should be formatted for easy reading and understanding.

2.5 Use of external resources

You are allowed but not at all required to consult resources outside of what is presented in the course. If you use an external resource in a significant way (not just googling an error message or a command that you forgot) please include a citation in your report. You are welcome to use features of python that were not shown in this class but you must understand fully what the feature does. The instructor reserves the right to ask you to explain any fragment of your code. An inability to do so may result in significant grade reduction or even more extreme consequences.

2.6 Collaboration

While students are allowed (and even encouraged) to collaborate and discuss projects the final submission must be the work of solely the submitting individual. Any assignment that is suspected of not being the student's own work will receive a zero and further disciplinary actions may also follow if they are deemed necessary. Any use of generative AI (e.g., ChatGPT) is prohibited in this class and will be considered a violation of UB's academic integrity policy.