CMPT 423-820 MINI PROJECT 3: CLUSTERING DIGITS

Team formation due date: March 21, 2025 at 4pm CST
Project due date: April 2, 2025 at 4pm CST

In this project, you will work in teams of three to four to implement and apply the K-Means algorithm to cluster the MNIST dataset and analyze the resulting clustering.

Team formation

In the first phase of the project, you will have to form teams of three or four students. You can form the same teams as for Mini Project 2. These teams must be submitted by submitting the *Mini Project 3 Team Formation* assignment on Canvas before the team formation due date.

Team Gitlab repository. For this project, each team must create its own private team repository. This can be done by forking the template repository found in the course Gitlab group here (need to insert link). This fork must be created by one team member and this team member must then invite all other team members and the instructor (Lucas Lehnert) as a "Maintainer" to this repository. The link to the repository location must be submitted on Canvas as the solution to the *Mini Project 3 Team Formation*. Only the team members and the instructor are allowed to have read access to the repository.

Late day policy. You can use late days for this project. If a late day is used, then a late day will be deducted from every team member's account. If one team member has only three late days remaining, but another team member has still four late days remaining, then the whole team can use at most four late days. However, if all team members have only three or fewer late days in their account, then the team can use at most three late days for this project.

Project instructions

To complete this project, follow the following steps:

1. Implement the K-Means clustering algorithm in the provided code base.
   (a) In your implementation, maximize the use of tensor arithmetic and only use for loops to iterate over each improvement iteration, the k different means, or the number of repeats.
   (b) In your implementation, do not use a the SciPy library. You can implement K-Means clustering entirely in numpy.
2. Run K-Means clustering for five different K values and for five different repeats.
   (a) Save each clustering result into a separate file. You do not need to submit these clustering result files. Even if your implementation is performant, it may need minutes to run throug for one clustering result.
3. Analyze the clustering in a one-page report.
   (a) In a line plot, show how distortion decreases as the K hyper-parameter increases. As discussed in class, distortion is defined as

$$J(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_k; \boldsymbol{Z}) = \sum_{i=1}^{N} \left|\left| \boldsymbol{x}_i - \sum_{k=1}^{K} z_{i,k} \boldsymbol{\mu}_k \right|\right|^2 . \tag{1}$$

   (b) Visualize the cluster centroids for the highest and lowest k. How does the k parameter influence finding all 10 digit classes? Does each cluster contain images of one or multiple digits?
   (c) To what extend can K-Means clustering be used to find the 10 digit classes in the MNIST dataset? Discuss your findings in a short paragraph at the end of your report.

Project report formatting instructions. The project report must be written in Latex using the provided template. The report must be at most two pages long, including figures but excluding any references you may want to use. This report must be submitted as a `report.pdf` file stored in the private team repository. This project report must clearly specify the team members in the author section. *Make sure to use your full name as listed in Canvas.*

Project implementation instructions. Similar to Mini Project 1, follow closely the provided method and function documentation to complete your K-Means implementation. Your implementation should be correct for any size and shaped input, as stated in the documentation stubs. To complete your implementation, replace all "`# insert your code here`" segments.

The clustering implementation help text with:

```
$ python -m kmeans.kmeans --help
Usage: python -m kmeans.kmeans [OPTIONS]

Options:
  --filename TEXT           Cluster result filename.
  --k INTEGER               K parameter.
  --max-iterations INTEGER  Maximum number of iterations.
  --epsilon FLOAT           Minimum distortion improvement per iteration.
  --repeats INTEGER         Number of K-Means clustering repeats.
  --help                    Show this message and exit.
```

Project deliverables

1. Implementation of the K-Means algorithm in the private team gitlab repsitory.
2. Project report also stored in the private team gitlab repository.

You do not need to submit the K-Means clustering result files. You are encouraged to include Jupyter notebooks used for creating figures for your project report, but the project report itself will be evaluated.

Project evaluation

The project will be evaluated based on the following aspects:

Grade weight	Aspect
60%	Correct implementation of the K-Means algorithm using only numpy functions and tensor arithmetic.
10%	Correct plot illustrating how distortion decreases as the K parameter increases.
10%	Correct figures illustrating the found cluster centroids.
20%	Discussion of your results and figures show to what extend the K-Means clustering algorithm can be used to correctly cluster MNIST digits.

The same grade will be assigned to each team member.