3945 – Advanced Machine Learning, Semester A 2026
Home Assignment 1: Exploring t-SNE for Dimensionality Reduction and Visualization

Lecturer: Dr. Leon Anavy
TA: Roei Arpaly

This assignment focuses on exploring the **t-Distributed Stochastic Neighbor Embedding (t-SNE)** technique, a widely used method for dimensionality reduction and visualization of complex datasets.

You will design, implement, and demonstrate the use of t-SNE while comparing your results to existing implementations and extending its functionality to handle new data points. This assignment is designed to deepen your understanding of dimensionality reduction techniques and enhance your skills in developing algorithms.
Note that some of the methods described in this assignment are for pedagogical purposes only and real-world implementation will require fine tuning and optimization to be useful and scalable.

Submission Guidelines

- Submit a single Jupyter notebook (.ipynb) that includes:
  - Appropriately formatted cells (markdown) describing the design, implementation, and showing any calculations.
  - All code and outputs.
  - Explanations for each step.
  - Observations and conclusions for each section.
- Ensure your notebook runs from start to finish without errors.
- Ensure that your notebook runs in under 5 minutes, end-to-end. If you are struggling with the runtimes, you can reduce runtime complexity by implementing vectorized functions, using sampling, or generating pipelines. Make sure to describe these choices, limitations, and any experiments you performed to validate your choices. For issues, reach out to course staff.

You may use external libraries as needed. Specify all dependencies in the notebook.

Grading

- The work will be assessed based on methodology, execution, and presentation.
- All figures and plots should be correctly labeled.
- Grading will be based on correctness, elegance of solution, and style (comments naming conventions, etc.).

Part 1: Design the Algorithm

- Describe the design you will use to implement t-SNE. Clearly describe the theoretical and practical considerations for the algorithm, any limitations, and discuss use-cases.
- Describe the algorithm, the various hyperparameters, and optimization strategy.

Part 2: Implement the Algorithm

- Based on your design, implement the t-SNE algorithm.
- Write **efficient**, **vectorized** code whenever possible.
- You must explain and justify the design.
- Handle hyperparameters correctly.

Part 3: Demonstrate the Algorithm

- Apply your implementation to a dataset of your choice. Clearly explain why t-SNE is or is not an appropriate algorithm for this dataset.
- Preprocess the data and explain the steps taken to prepare it for t-SNE.
- In past years, the MNIST dataset was used in this course. You are welcome to use MNIST if you follow the above guidelines.

Part 4: Compare to Existing Implementations

- Apply the built-in t-SNE algorithm from scikit-learn to the same dataset.
- Compare your implementation to the library implementation in terms of visual results, computational performance, and quality of embeddings. Justify your findings.

Part 5: Mapping New Data Points

- Devise and implement a method to extend your t-SNE algorithm for embedding new data points into a precomputed 2D map. Your method should consider the positions of the original training data and their transformations. Explain your method.
- Test and demonstrate your method on a subset of your dataset and explain the results.
- Propose a performance measure for this extension and use it to assess your method. Explain your proposed measure.

Experimentation Guidelines

- Clearly describe the experimental setup, including:
  - Dataset characteristics.
  - Data preprocessing steps.
  - Parameter selection for t-SNE.
  - Justification for key design and implementation choices.
- Use appropriate plots to demonstrate results.
- Discuss the observations and insights derived from the results.

Generative AI Report

- Compile a report on your use of Generative AI tools, include details on at which stage you used the tools and for what purpose, and what code was generated with the tools.

Ensure all submitted work is original and that you fully understand it. You may be asked to explain your work in an in-person review.

Pair Work: Submit in pairs, with one submission per pair.