Embedding: Self Organizing Maps (SOMs) and Locally Linear Embedding (LLE)

Consider first solving the part on LLE, then circle back to SOMs.

Exercise 9.1: 1d Self-Organizing Map for 2d data (3 points)

(a) Generate $p = 1000$ data points that are uniformly distributed inside the rectangle that is $\underline{\mathbf{x}} \in [0, 2] \times [0, 1]$

(b) Implement a one-dimensional Self-Organizing Map (Kohonen network, online algorithm for SOMs) using a Gaussian neighborhood function

$$h_{qp} = \exp\left(-\frac{(q-p)^2}{2\sigma^2}\right)$$

(c) Fit different maps with $M \in \{4, 8, 16, 32, 64, 128\}$ nodes (prototypes) to the data.

Hint: Anneal both the learning rate ε and the neighborhood width σ. The start value σ_0 has to be large enough to unfold the randomly initialized (scrambled) map in the first iterations. The learning rate plateau at ε_0 should last until the neighborhood width has decayed by a substantial amount. The learning rate should then decay inversely with the iterations.

(d) Plot the final map in the original data space, i.e. the locations of the prototypes and how they are connected to one another. Do so for each number of nodes M separately.

Exercise 9.2: 1d Self-Organizing Maps for 3d data (2 points)

(a) Download and visualize the data contained in the file `spiral.csv`. It contains data described by three coordinates x,y,z.

(b) Adapt/reuse your implementation for SOM to fit one dimensional maps with $M \in \{16, 32, 64, 128\}$ nodes to this dataset.

(c) Initialize the prototypes of your map as a chain with equally spaced links along the z axis, i.e. with $x = 0$, $y = 0$, and $z = 0, \ldots, 5$.

(d) Plot the final prototypes of the map in the data space (for each value for M separately)

Exercise 9.3: 2d Self-Organizing Maps for 3d data (3 points)

(a) Visualize the 3d-data in the file `bowl.csv`.

(b) Extend your SOM implementation to fit two-dimensional maps with an $M \times M$ (cartesian) grid topology to this dataset. Set $M \in \{8, 16, 32\}$ as far as your computing resources allow it. Extend your neighborhood function accordingly (i.e. $h_{qp} \rightarrow h_{\underline{\mathbf{q}}\underline{\mathbf{p}}}$).

(c) Experiment with different ways for initializing the prototypes:

  (i) randomly,

  (ii) in an *informed* way, e.g. arrange the initial prototypes as small grid centered on the data mean and spread along the first 2 principal directions of the data.

(d) Plot the map in data space (protoype locations and their "connections") at

  (i) t_0,

  (ii) at some intermediate iteration, and

  (iii) in its final configuration.

Exercise 9.4: Locally Linear Embedding (2 points)

1. Find an off-the-shelf implementation for Locally Linear Embedding (e.g. Scikit-Learn).
2. Apply LLE to the toy data sets above using:
   - $M = 1$ embedding dimensions for the spiral data,
   - $M = 1$ and $M = 2$ embedding dimensions for the bowl data.
3. Plot the data points in embedding space using an arbitrary color scheme (required for the next step; see LLE lecture slides for an example from Roweis & Saul).
4. Plot the data points in data space using the same colors as in embedding space above to indicate the distance of the data points within the obtained low-dimensional manifold.

Total 10 points.