HY673 - Spring 2026 Problem Set #2
Yannis Pantazis (17/03/2026)

Instructions

- **Due date:** Wednesday April 1st, 2026
- Submission via e-mail to the class account: hy673@csd.uoc.gr
- Provide one file with the written solutions.
- Provide one folder with code.
  - The name of each file in the folder should indicate the respective exercise.
  - Your code should run on Colab.
- All assignments in this course are individual, not group assignments. You may freely discuss homework assignments with your classmates. The final solutions, however, must be written entirely on your own.
- You are allowed to use generative AI tools (e.g., ChatGPT, Gemini) only for grammatical corrections unless explicitly permitted otherwise. To maintain academic integrity, students must disclose any use of AI-generated material.

Problem 1 (Gaussian Mixture Modeling on Protein Embeddings – 20 points)**.** *We consider protein representations derived from the dataset* `https://huggingface.co/datasets/pitagoras-alves/swissprot_protein_embeddings-esm2`, *which contains precomputed embeddings from the ESM-2 protein language model. Use the file* `emb.esm2_t6.parquet` *which provides one embedding vector per protein. Such embeddings capture structural and functional properties of proteins learned from large-scale sequence data. The goal is to model the distribution of protein embeddings using a Gaussian Mixture Model (GMM) with diagonal covariance matrices and analyze its structure.*

(a) Data loading and preprocessing.

Load the dataset from the parquet file and construct a data matrix $X \in \mathbb{R}^{N\times d}$. Extract the embedding vectors and stack them into a matrix and subsample the dataset to $N \approx 5000$ for computational efficiency.

(b) Gaussian Mixture Model with diagonal covariance.

Assume the data are generated from a mixture of K Gaussian distributions:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k),$$

where each covariance matrix is diagonal:

$$\Sigma_k = \mathrm{diag}(\sigma_{k1}^2, \ldots, \sigma_{kd}^2).$$

Fit a GMM using the EM algorithm with the number of Gaussian components, $K = 10$. Explain why diagonal covariance is computationally advantageous in high dimensions. Repeat the estimation with different initialization of the parameters. Comment on how similar are the obtained results.

(c) Model selection via Bayesian Information Criterion (BIC).
To determine the optimal number of components, use the BIC criterion:

$$\mathrm{BIC} = -2 \log \hat{L} + p \log N,$$

where $\hat{L}$ is the likelihood for the estimated parameters and p is the total number of model parameters. Compute the BIC for $K = 1, \ldots, 20$ and plot it as a function of K. Select the optimal number of components K^*. Derive the total number of parameters p for a diagonal GMM in dimension d.

(d) Visualization via dimensionality reduction.
Since the data are high-dimensional, project them to 2D using either t-SNE or UMAP. Visualize the embeddings in 2D. Color each point according to its assigned GMM cluster. Add to the visualization the contour of the fitted GMM. Compare the structures revealed by t-SNE and UMAP. Comment on whether clusters appear well-separated and interpretable.

Problem 2 (Conditional generation of names – 30 points)**.** *Extend the autoregressive (AR) model from Tutorial 5 to conditional name generation. Encode each country as a one-hot vector and concatenate it with the input at each time step of the RNN. Implement a model that, given a country as input, generates a list of 10 plausible names. Repeat the experiment using both LSTM and GRU architectures.*

Problem 3 (Maximum number in a sequence – 30 points)**.** *(a) Create a dataset consisting of input sequences of integers and their corresponding maximum value. Each sample is of the form*

$$(x_1, x_2, \ldots, x_L)$$

with target

$$\max(x_1, x_2, \ldots, x_L).$$

Generate 10^5 samples where L is randomly chosen in the range $[5, 100]$ and the integers are uniformly drawn from $[0, 10^6]$.

(b) Train a Transformer model that receives the sequence $(x_1, \ldots, x_L)$ and directly outputs the maximum value. Plot the training loss as a function of iterations and report the accuracy on a held-out test set.

*(c) Train a second Transformer model that outputs the solution using **intermediate steps**. The model should simulate a pairwise comparison procedure. For example, $[7, 2, 9, 4]$*

could produce an output sequence such as $[7, 9, 4] \to [9, 4] \to 9$*, with the final token corresponds to the maximum element. Plot the training loss and report the accuracy on the test set.*

(d) Compare the two approaches: 1. direct prediction and 2. prediction with intermediate reasoning steps. Discuss differences in: (i) convergence speed, (ii) accuracy, (3) generalization to longer sequences (e.g. $L = 200$*) and (iv) integers larger than the upper value,* 10^6*.*

Problem 4 (Sorting a sequence – 30 points)**.** *(a) Create a dataset consisting of integer sequences and their sorted version. Each sample is of the form*

$$(x_1, x_2, \ldots, x_L)$$

with target

$$\text{sort}(x_1, x_2, \ldots, x_L).$$

Generate 10^5 *samples where* L *is randomly chosen in the range* $[5, 100]$ *and the integers are uniformly drawn from* $[0, 10^6]$*.*

(b) Train a Transformer model that receives the sequence $(x_1, \ldots, x_L)$ *and directly outputs the sorted sequence. Plot the training loss as a function of iterations and report the accuracy on a held-out test set.*

(c) Train a second Transformer model that outputs the solution using ***intermediate steps****. The model should simulate a pairwise comparison procedure similar to a sorting algorithm (e.g., bubble sort or merge-style comparisons). For example,*

$$[7, 2, 9, 4]$$

could produce intermediate outputs such as

$$[2, 7, 9, 4] \to [2, 7, 4, 9] \to [2, 4, 7, 9].$$

The final output corresponds to the sorted sequence. Plot the training loss and report the accuracy.

(d) Compare the two approaches: 1. direct prediction and 2. prediction with intermediate reasoning steps. Discuss differences in: (i) convergence speed, (ii) accuracy, (3) generalization to longer sequences (e.g. $L = 200$*) and (iv) integers larger than the upper value,* 10^6*.*

(e) Visualize the attention weights of the trained Transformer. Plot attention maps for several heads and layers for representative sequences. Discuss whether the attention heads appear to discover comparison patterns (e.g., attending to smaller or larger elements during sorting).