Homework 2
IST 597
Physics-Informed Machine Learning

Subarna Pudasaini (sfp5828@psu.edu)

Question 1

Implement Levy and Lindenbaum's streaming SVD algorithm and deploy it for the MNIST dataset where images are streamed in batches of 50. Levy and Lindenbaum use smaller QR and SVD decompositions - use prebuilt libraries (such as np.linalg.) for these.

Ans:

Code: hw2.ipynb

Code from scratch: from_scratch.py

The final SVDs (first 3) for the forget-factor of 1 are given below:

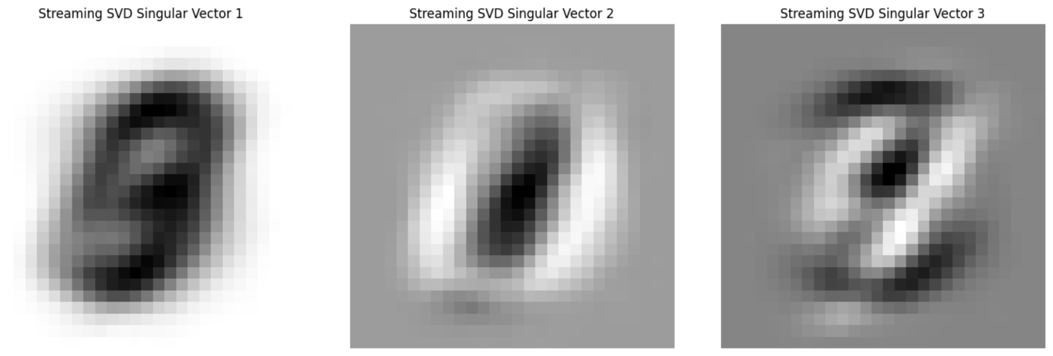

Figure 1: SVD after processing all batches

Question 2

Derive the time-computational complexity of Golub-Reinsch for computing the SVD.

Ans:

The major steps in the Golub-Reinsch algorithm are:

1. Bidiagonalization using Householder Transformations

   The bidiagonalization process requires matrix-matrix-matrix multiplication on the matrix's left and right sides.

   (a) $B_1 = H_1 A$

   Let A be of size m x n. Then, the size of H_1 is n x n.

   Hence, the computational complexity for the matrix-matrix multiplication is $\mathcal{O}(m^2 n)$.

   (b) $B = B_1 H_2$

   The size of B_1 is m x n. Then, the size of H_2 is n x n.

   Hence, the computational complexity for the matrix-matrix multiplication is $\mathcal{O}(n^2 m)$.

Therefore, the total complexity of the bidiagonalization process is $\mathcal{O}(m^2n) + \mathcal{O}(n^2m)$.

2. Diagonalization of B using QR Algorithm

   This involves performing QR decomposition iteratively on the bidiagonal matrix B.

   (a) For the bidiagonal matrix B, perform QR decomposition at each iteration:

   $$B_k = Q_k R_k$$

   where Q_k is an orthogonal matrix and R_k is an upper triangular matrix.

   (b) Update the matrix for the next iteration:

   $$B_{k+1} = R_k Q_k$$

   The computational complexity for each QR iteration is $\mathcal{O}(n^3)$.

   So, the total complexity for the diagonalization step is therefore $\mathcal{O}(t \cdot n^3)$, where t is the number of iterations until convergence.

Since the bidiagonalization and diagonalization processes dominate the overall computational complexity of the Golub-Reinsch algorithm, the computational complexity of the algorithm is:

$$\mathcal{O}(m^2n) + \mathcal{O}(n^2m) + \mathcal{O}(t \cdot n^3)$$

Question 3

Assess run-times for varying batch-size in a plot with x-axis as batch size and y-axis as average run-time for 10 trials. Assume a forget-factor of 1.0 and visualize the evolution of the first three singular vectors with each batch as a video (mp4/gif) and upload to Canvas.

Compare the singular vectors obtained by your implementation of Levy and Lindenbaum (after processing all batches) to np.linalg.svd qualitatively for different values of the forget factor.

Ans:

All the figures are also uploaded to the canvas.

Runtime vs. Batch-size

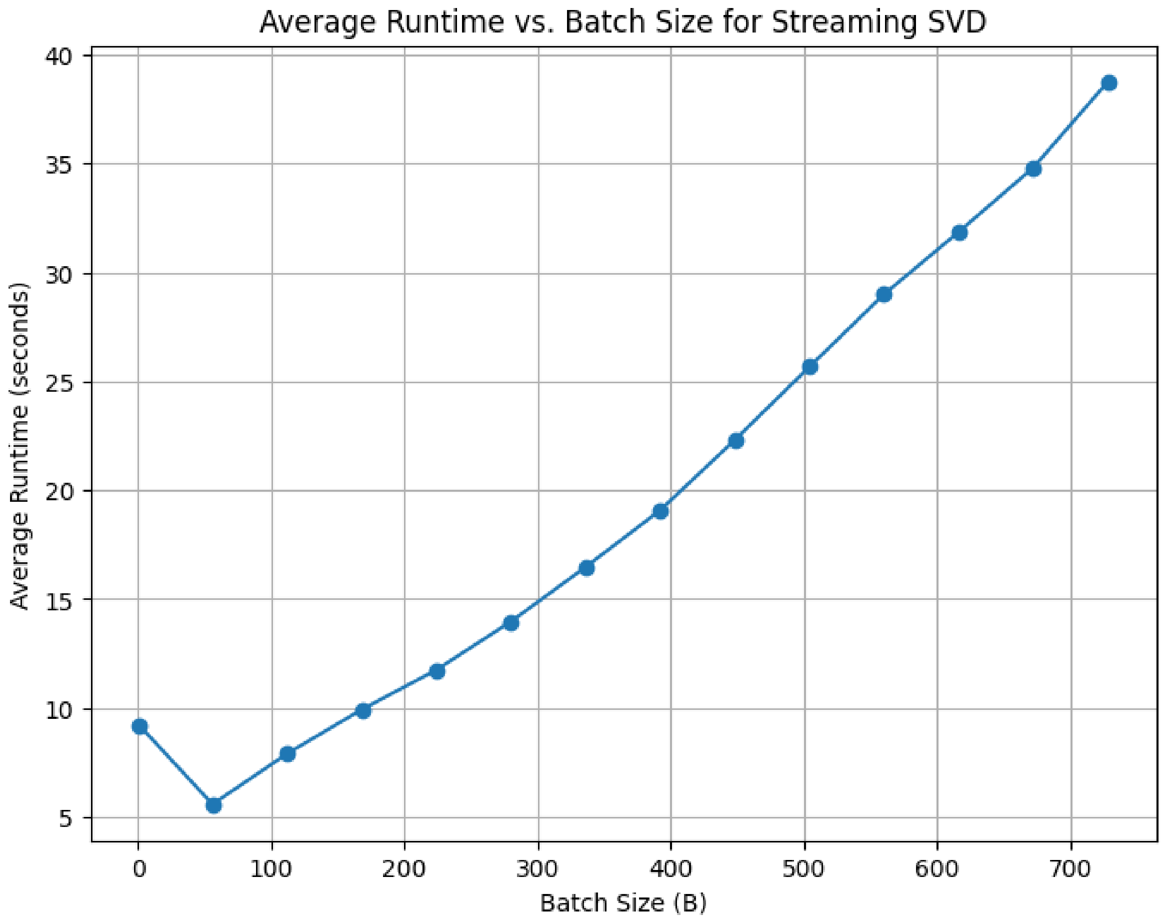

Figure 2: Runtime vs. Batch-size

Comparison of singular vectors: Levy and Lindenbaum vs. np.linalg.svd

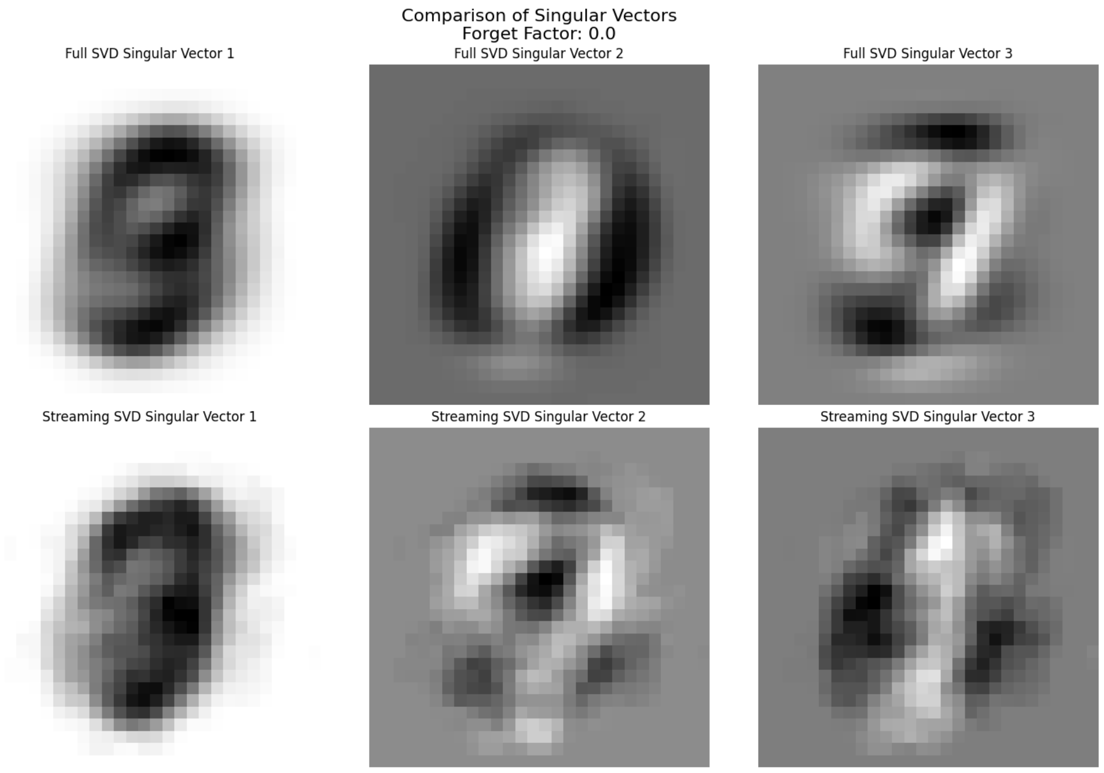

Figure 3: SVD comparison; forget factor = 0

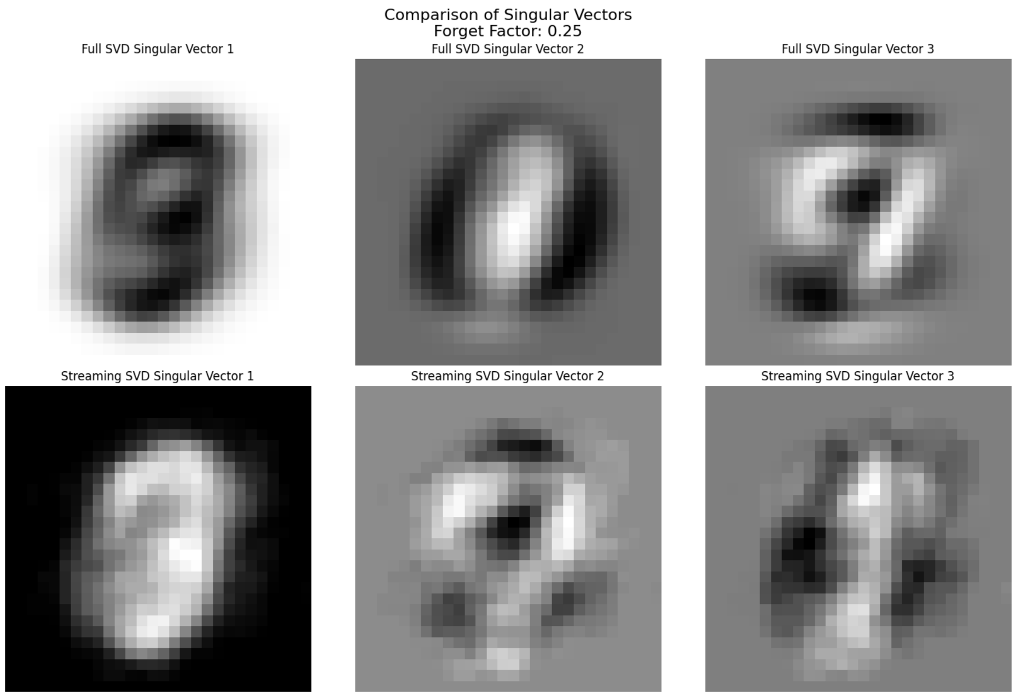

Figure 4: SVD comparison; forget factor = 0.25

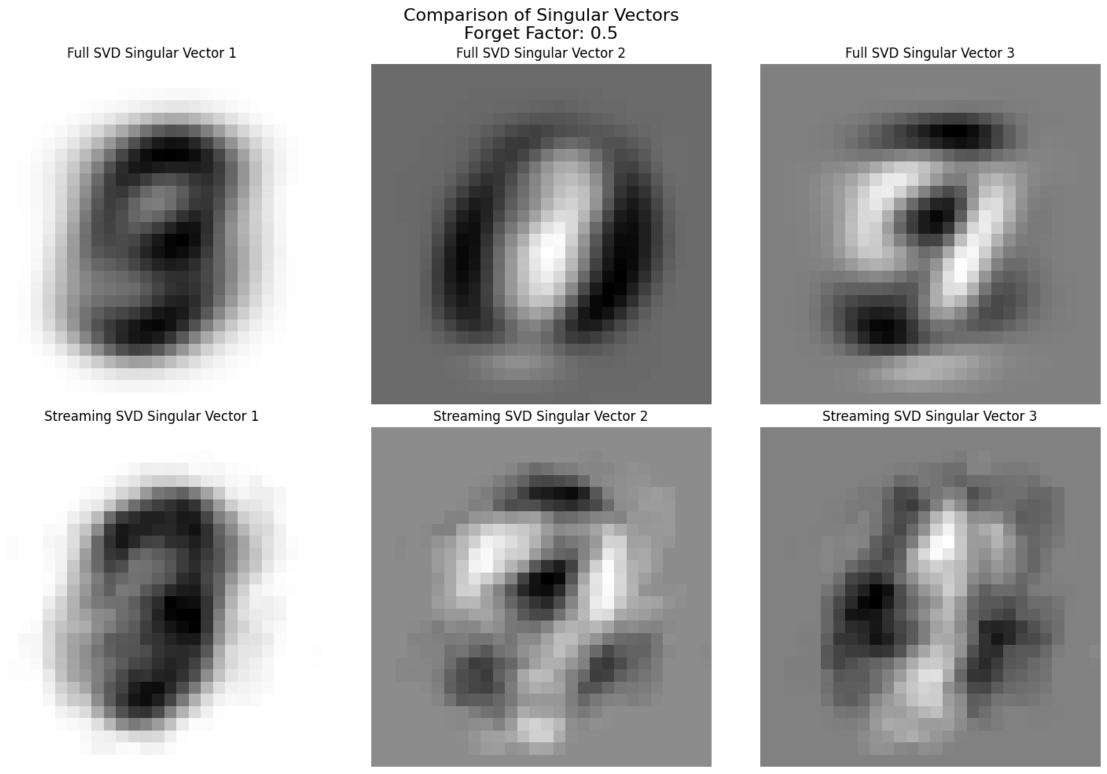

Figure 5: SVD comparison; forget factor = 0.5

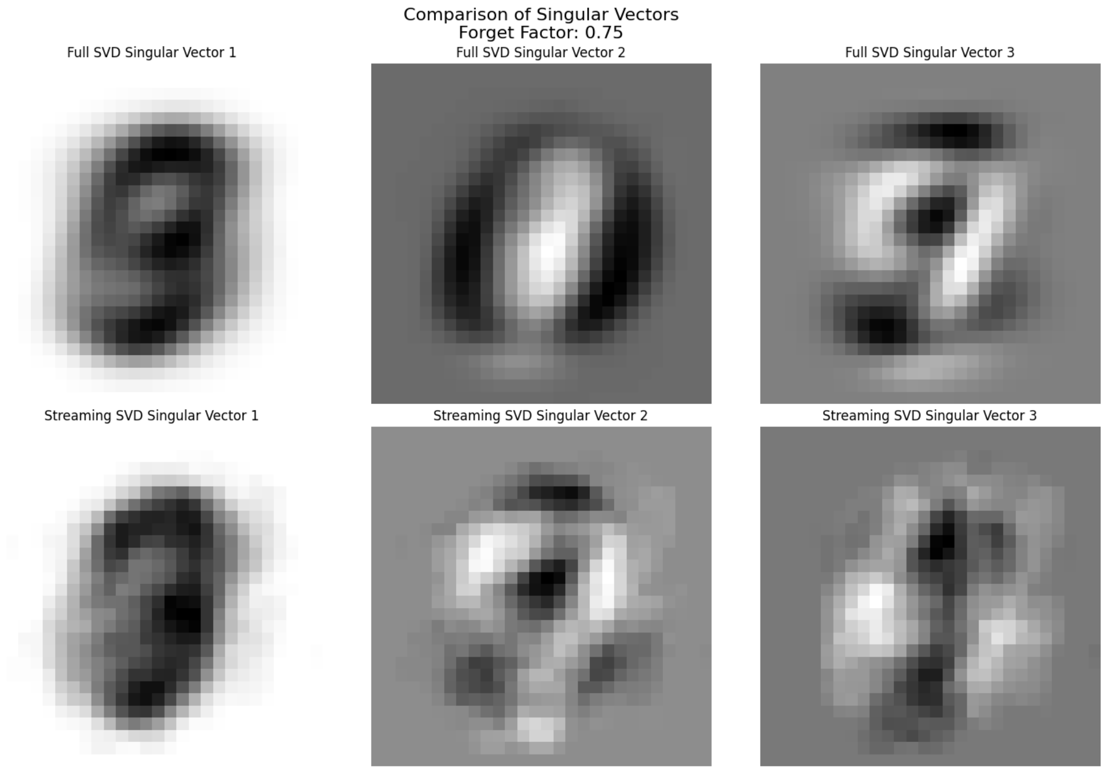

Figure 6: SVD comparison; forget factor = 0.75

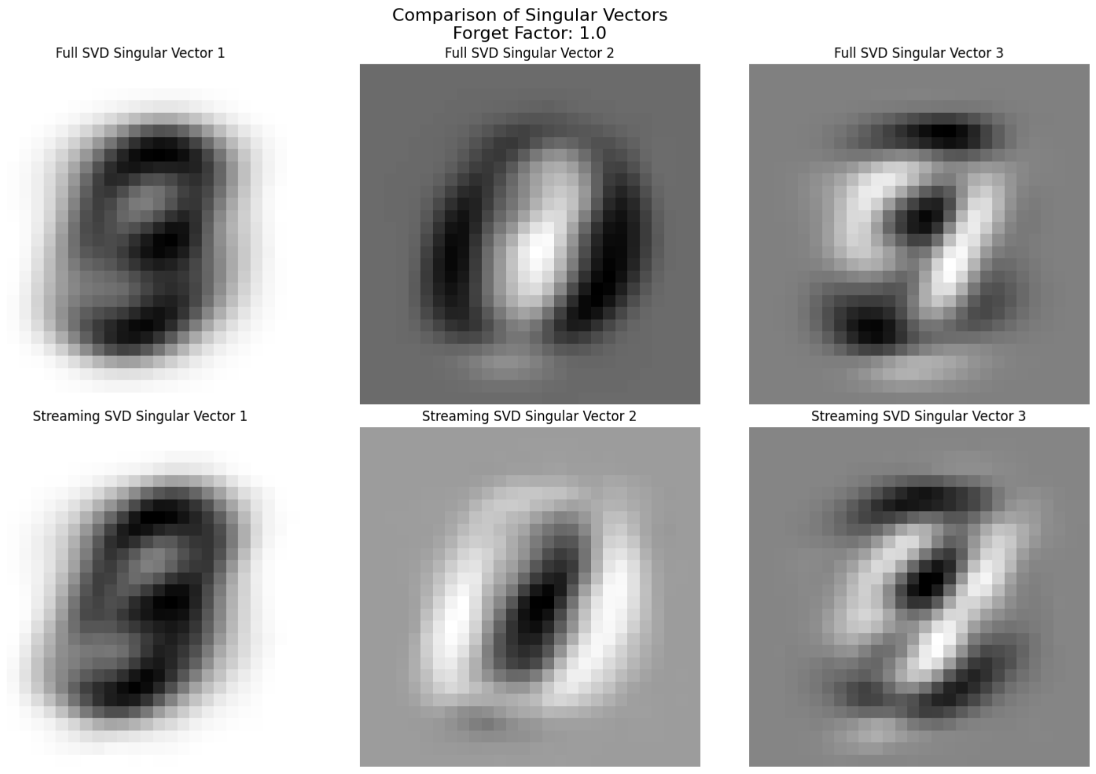

Figure 7: SVD comparison; forget factor = 1.0