Data-Driven Techniques: 2^{nd} Homework

Problem 2.1: In the file `data21.mat` you can find a generative model for the generation of the "handwritten" numeral eight (8). The file contains two matrices A_1, A_2 and two vectors B_1, B_2. Matrix A_1 has dimensions 128×10, while A_2 dimensions 784×128. Vector B_1 is 128×1 and B_2 is 784×1. The neural network is fully connected and in the hidden layer we apply the ReLU for activation while in the output the sigmoid. Input Z has dimensions 10×1 and its elements are i.i.d. realizations of a Gaussian with mean 0 and variance 1. The output is a vector of length 784×1 which we use to form an image 28×28 with a "handwritten" 8. The equations leading to the output are the following

$$\begin{aligned} W_1 &= A_1 * Z + B_1 \\ Z_1 &= \max\{W_1, 0\} \text{ (ReLU)} \\ W_2 &= A_2 * Z_1 + B_2 \\ X &= 1./\big(1 + \exp(W_2)\big) \text{ (Sigmoid).} \end{aligned} \tag{1}$$

We recall that ReLU is applied to every element of W_1 and similarly the sigmoid applied to every element of W_2. ***Be careful to use this specific form of sigmoid otherwise your output will not be correct***. The output X is a vector 784×1 that contains the *columns* of the image stacked one after the other. If for example you generate a realization of Z and apply the previous equations then in Matlab you can change the output vector into the right matrix by applying the command $X_{2D} = \texttt{reshape}(X, 28, 28)$. You may also admire your results using the command $\texttt{imshow}(X_{2D})$. Generate 100 realizations of Z and apply the generator giving rise to 100 different 8. Place the generated 8s in a single image arranging them in a table of $10 \times 10 (= 100)$. Include this as figure in your report.

Problem 2.2: The generative model of Problem 2.1 is employed in this and the next problem and is used to capture the statistical behavior of a "handwritten" 8 which is a point in a space of dimension 784 but lives in a manifold of dimension 10 (the size of the input). In this problem we will use the generative model to do inpainting, namely to recover an image from a part of it. In the file `data22.mat` you will find two matrices X_i and X_n of dimensions 784×4. X_i contains four 8s (in vector form) which are the ideal images. This information **you are not allowed to use for any form of computation**, they exist only for comparisons to verify whether your processing is successful. Matrix X_n contains the X_i but we have also added noise with mean 0 and unknown variance. The columns of X_n is the measurements you are given. **Attention!!!** do not use the whole length of each column but only from the 1st to the N-th element. Elements from $N+1$ to 784 are considered lost or so noisy that are completely useless. We would like to recover the missing part (elements $N+1$ to 784) removing at the same time the additive noise in the whole vector. We will process the first N elements of each column of X_n in order to recover the whole column of length 784. This processing must be applied to each column separately. Try different values for N. Start with $N = 500$ then 400, 350, 300, and discover at which point the recovery fails.

Problem 2.3 In the file `data23.mat` you will find two matrices X_i and X_n. The first (same as in the previous problem) has dimensions 784×4. X_i contains four 8 (in vector form) which are the ideal version. This data **you cannot use for any form of processing**. You can use them only to check whether your processing was successful. Matrix X_n has dimensions 49×4 and was generated by resolution reduction of the ideal versions X_i and by adding noise. Specifically each column of X_i becomes an image of dimensions 28×28 and then divided into a grid. Each grid square has dimensions 4×4 and the corresponding 16 pixels are replaced by **a single pixel** with value the average of the values of the 16 pixels. This way we generate an image of dimensions 7×7 in which we also apply additive noise with mean zero and unknown variance. Next the image is transformed into a vector of length $49 = 7 \times 7$ by stacking the columns one after the other. This process was applied to each of the four columns of the ideal matrix X_i (corresponding to different 8s) and this is how matrix X_n is generated containing the measurements you must process. Processing must be applied to each column of X_n separately in order to recover the corresponding column of the ideal matrix X_i (or something close). **We observe that the images we will process have resolution which is 16 times smaller than the resolution of the ideal images!!!!**

Hints

The transformation applied to both Problem 2.2 and 2.3 is **linear**. Consequently our data (columns of matrix X_n) are of the form

$$X_n = T\,X_i + \text{noise}$$

where T appropriate matrix. In case of Problem 2.2 matrix T has dimensions $N \times 784$ and is of the form $T = [I\ 0]$ where I is the identity matrix of size $N \times N$ and 0 a zero matrix of size $N \times (784 - N)$. Indeed if you multiply T with X then TX isolates the first N elements of the vector X. In Problem 2.3 matrix T has size 49×784. The first row of T when multiplied with an image 28×28 (after it is transformed into a vector 784×1) must return the value of the pixel (1,1) of the low resolution 7×7 image. Therefore the first row in positions 1 to 4 it is equal to $\frac{1}{16}$, this is also true for the positions 1*28+(1 to 4), $2 * 28$+(1 to 4), $3 * 28$+(1 to 4), the remaining elements of the first row are 0. The second row of T which corresponds to element (2,1) of the 7×7 image will have at positions 5 to 8 the value $\frac{1}{16}$ and also at positions $1 * 28$+(5 to 8), $2 * 28$+(5 to 8), $3 * 28$+(5 to 8). Similarly you need to discover the other rows of T which correspond to each (i, j) pixel of the low resolution 7×7 image.

According to the optimization method introduced in Lecture 09 you will apply GD to estimate the appropriate input Z to the generative model given to you in the file `data21.mat`. Characteristic example of the reconstruction result is

for Problem 2.2 with $N = 400$. Note the black orthogonal in the middle image corresponding to the missing pixels (384 lost out of a total of 784) also observe the presence of noise. The left eight is the ideal (1st column of X_i) and the eight on the right is the reconstructed. Observe that the noise is gone and the missing part has been recovered quite accurately! You need to obtain similar results with your own processing for both cases of the transformation (Problems 2.2 and 2.3). In fact the results in Problem 2.3 are more impressive since you process 49 pixels instead of the 784 of the ideal image!! *In Problem 2.3, for better presentation, the low resolution 7×7 images should be increased to the same size 28×28 as the ideal and the final images. You can do this by replacing each pixel with a square of size 4×4 with all 16 pixels having the same value as the original pixel they replace. Of course the resolution is still low but you have an image of the same size as the other two.*

Attention!

- Please submit your report **ONLY if you are taking the class for credit and your name appears in Progress**. Reports with names not appearing in Progress will not be evaluated.

- You must upload your report to E-class by Friday 6 December, **BEFORE MIDNIGHT**. **There will be absolutely no extension to this deadline.**

- The ONLY acceptable file type is PDF. Name your file as follows: firstname-lastname-2.pdf

- In the first page of your report write your full name and university code (Arithmos Mitroou) legibly.

- DO NOT send Python or Matlab code in separate files. Include your code in the pdf file as text **AFTER the presentation of your results of ALL the problems**. Code only, is not considered presentation of results and if your report contains only code then your grade will be 0.

- Do not simply report final outcomes. Since you will be using gradient descent algorithms, present the learning curves where we can see how the algorithms converge.

- Please make sure your files are small to avoid communication problems because of excessive size.

- **DO NOT SEND YOUR REPORT OR PART OF YOUR REPORT DIRECTLY TO MY PERSONAL EMAIL OR PROF. PSARAKIS**, it will not be accepted. It is part of your training to learn how to comply with hard deadlines!!!