Project 2 - EEL 6814
Due November 30, 2023 --- Team of two students.

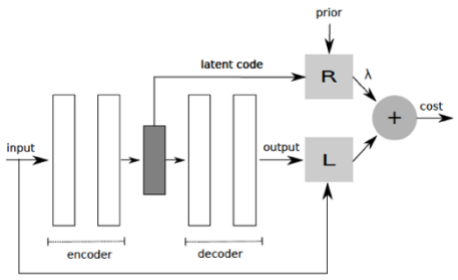
1- design a stacked autoencoder network (SAE)

SAE projects data to a subspace to obtain features (codes) that can be then used for classification. You will use again the Kuzushiji-MNIST from project 1. We will compare two different approaches: First train a SAE (5 hidden layers) with MSE/cross entropy as found in the literature. Once the SAE is trained, use the bottleneck layer outputs (features) as inputs to a Support Vector Machine (SVM) or MLP classifier. I suggest the SAE layers to be 800-200-XXX, i.e. the number of units XXX of the bottleneck layer (which selects the dimensionality of the feature space) is selected by you. Present your strategy to find the best size of the bottleneck layer (plus the other hyper parameters) and support it with experiments.
Use the MLP classifier results from project 1 as a comparison to evaluate the obtained results. Compare the network sizes of each approach (MLP versus SAE+Classifier) and correlate the performance drop with the size of the bottleneck layer using confusion matrices. Explain the results.

Add impulsive noise to the input image (alter the image pixels with a probability of 10% using a mid-range grey color since the images are mostly black and white) and present the noiseless image as the desired response to train the same SAE. Substitute the MSE by correntropy cost with a small kernel size (need to validate it) and show that it cleans better the noise from the output. Quantify the effect of added noise on the codes themselves, by comparing classification accuracy with the previous SAE+ Classifier and the versions trained with MSE and with correntropy. Explain results from first principles.

2- design a penalty function that enhances discrimination in the latent space.

With this goal in mind, we are going to add a regularizer to the SAE reconstruction cost function, based on ideas of distances. Since this a classification problem, we know the number of classes, so we can create a set of targets (a prior) in the space of the codes that should be maximally discriminative i.e. a "constellation", as done in communication systems, to help organize the codes when we train SAE using reconstruction error as the cost. So, we are using more information to train the SAE, and discrimination of codes for classification should improve. To simplify the task, since during training we know the class of each input, we can measure the distance of the current code to its constellation target, with the goal of minimizing their distance as an added cost constraint. To implement this idea, we will create a cost function (R in the figure) as $R = \frac{1}{K}\sum_{i=1}^{K} d_i$, where K is the batch size and d is a distance between the current code and the current target. Then the new cost function for SAE is $J_{new} = L + \lambda R$, where L is the MSE reconstruction cost. This is a "mental model" that you must fill-in details with your team member.



The details are: the definition of the constellation targets, the size of the batch K, the distance measure you will use, the selection of λ, the order of presentation of samples and the integration with the training of the SAE. Evaluate R first in the training set to quantify the quality of the constraint. Then you will have to retrain the SAE with the new cost: use the same bottleneck size and classifier as in part 1 to compare the classification results. Validate selection of λ, R together in a 3D space. Explain the quality of the new results. Lastly, select XXX as 10 and use directly the codes as class assignments in the test set, and compare results with the previous cases.
Remember that the goal is to present a comprehensive comparison amongst all these ML algorithms, so use confusion matrices for each method, and quantify the computation time in your computer. The project was thought out for two students, so mention who did what. The report format is as in Project 1.