HOMEWORK HW4
Math 6365 // Spring 2018

1 Data Base and Main Expected Outputs

1. Select a Classification task for which the Data Base has at least 5000 cases and at least 500 descriptors per case.
Here are potential data sets of this type:
- small patches in 2D aerial or satellite images , to classify patch content into "urban", "farming", "forest" , "prairies" .
- intra-day current prices for 500 stocks (available at 1 min intervals), to decide if two specific target stocks will go up or down in the next 10 minutes.
This 2nd type of data can for instance be downloaded from yahoo.finance.

The first task below, before implementing automatic classification, is to generate , train , and compare auto-encoders.
The actual classification task will only then be studied by inserting an auto encoder into an MLP classifier.

2. Give a brief description of the data base and of the classification task :
- word description of generic "cases"
- description and sizes of the classes
- number "p" of descriptors per case
- range and nature (continuous or discrete) of each descriptor
- description of the number and types of expected outputs
- description and sizes of training set and test set

2 AutoEncoder (AE) Architecture

1. Define the following AUTO-ENCODER architectures based on the MLP paradigm.
Select MLP architectures with three layers $INPUT \to H \to OUTPUT$ of dimensions $[p, h, p]$.
The dimension "h" of the hidden layer H will have to take several tentative values for this project.

The output layer should IDEALLY (after training) generate an estimated output reconstructing as well as possible the current input.

2. Select a response function (either sigmoid or RELU). Indicate how you will encode the descriptors $D_1...D_p$ of each case on the INPUT nodes.
If you use the sigmoid response function each descriptor D_j should be replaced by

$$d_j = (D_j - m_j)/(M_j - m_j)$$

where m_j, M_j are respectively the minimum and the maximum of the values $D_j(k)$ over all the cases CASE(k) , with k = 1... N.

If you use the RELU response function each descriptor D_j should be replaced by

$$d_j = (D_j - m_j)$$

with m_j as above.

3. Select 3 exploratory values for h , namely $h1 < h2 < h3 < p$. To this end you can for instance perform a PCA analysis , compute and order the "PCA eigenvalues" in decreasing order, and select 3 cut-off energy level ratios $R1 = 50\%, R2 = 75\%, R3 = 90\%$. The corresponding numbers of ordered eigenvalues can then be h1, h2 h3.

3 Automatic learning for three AEs

1. Describe briefly the software tool ST you will use for standard MLP learning in order to minimize the mean squared error MSE on the training set with validation on the test set

2. Outline the options offered by the tool ST for MLP learning, for random weights initialization, batch size , learning rate $\epsilon(n)$, stopping rule, intermediary outputs, and then indicate your own possibly multiple choices for these options

3. For each one of the 3 AEs defined above , perform several runs of automatic learning with varying options and initializations, and for each AE

keep the best of these runs

4. For each learning run, call W(n) the successive values of the weights vector and RMSE(n) the Root Mean Squared Error computed on the current batch. Plot as functions of n the three quantities
$RMSE(n), ||W(n+1)-W(n)||, ||G(n)||$ where the gradient G(n) is computed by

$$G(n) = (W(n+1) - W(n))/\epsilon(n)$$

Interpret these three curves. For clearer interpretation, each $||V||$ for a long vector V should be normalized by the dimemsion of V.

5. For each learning run, compute *on the full training set* the "global performance" curve $r \rightarrow GRMSE(rN)$ where N is the number of cases in the training set , $r = 1, 2, 3, \ldots$ is the number of "epochs", and each GRMSE(rN) is the root mean square performance on the *whole* training set.
Perform the same operation for the test set. Plot these two "global performance" curves on the same plot as functions of the number "r" of epochs, and interpret comparatively these 2 graphs.

6. Analyze comparatively the performances of the 3 trained auto encoders and indicate how you select the best AE among these 3 AEs.

4 Activity analysis of hidden layer

For each one of the three AEs trained above, perform the following analysis.

1. Each one of the M existing input vectors X(k) generates a configuration Y(k) of the hidden layer H. Implement PCA for the cloud of M vectors Y(k).
Compute the number of PCA eigenvalues needed to achieve 90% of the energy for this cloud .

Assign a specific color to each class. Compute and display the projection of the colored classes onto the first three eigenvectors

2. For each node "i" of the hidden layer , identify the input vector $X(k^*)$ such that

$$Y_i(k^*) = \max_k Y_i(k)$$

The corresponding input $X(k^*)$ belongs to one specific class $C(i)$.
Attribute the color of class C(i) to the node "i". Then display graphically the colored layer H.
Compute how many nodes of H are attributed to each class.

3. Compute the average activity of node $i \in H$ by

$$act(i) = (1/M) \sum_{k=1}^{M} Y_i(k)$$

Re-order these h average activities act(1) ... act(h) in increasing order, and plot the reordered activity values.
Compute the 5 % quantile Q of these activities.
Identify all the nodes "i" which have "low activity" $act(i) \leq Q$.
Try to explain why these specific nodes have low activities.
Try to evaluate if these nodes could be removed without much impact on the output quality of the AE.

4. Perform the same analysis of low activity nodes within each class C, by defining
M(C) = size of C,
K(C) = set of all input vectors X(k) such that Case(k) is in class C

$$act_C(i) = (1/M(C)) \sum_{k \in K(C)} Y_i(k)$$

5. Compare the preceding results for the 3 trained AEs

5 Auto-Encoder insertion in MLP classifiers

1. Let AE be the "best" one of the 3 auto-encoders just trained.
Let INPUT $\rightarrow$ H be the first two layers of AE , and FIX completely the already computed weights and thresholds linking INPUT to H.
Define an MLP classifier with 4 layers

$$INPUT \rightarrow H \rightarrow HH \rightarrow CL$$

where
- HH is of size hh to be selected by trial and error

- CL has exactly one node per class
- the weights and thresholds from H to HH and from HH to CL are to be learned by automatic learning

Try different sizes hh for HH. For each tentative size hh , implement MLP classification learning on the training set for this MLP (while keeping frozen the weights linking INPUT to H).

2. Compute the performances on the training set and on the test set, and use these numbers to select the "best" size hh.
Compute and display the confusion matrices on training set and test set for the best "hh" .