Homework 3 - Machine Learning

Authors: Manuel di Pietro, Lorenzo Pichilli

We implemented, using **Python 3**, a procedure to create the MLP architecture for general learning as described in Lecture 14 for the prediction of any set of feature variables. The procedure includes the use of a function that encodes/creates a given dataset as needed for the MLP architecture.

To encode our data set we have implemented `CharacteristicVector` class that encodes each column in a characteristic vector, considering each number like a symbol. If the number is a float, we round it to get the integer value.

Then each column is encoded in a characteristic vector with ***x*** bits, depending on the number of classes defined by the `CharacteristicVector` class attached to its. For example, if a column contains the 0, 1, 2, 7, 8 and 15 values, then the encoded version has 6 bits; if a row has that column value at 15, then its encoded value is 000001, where the bit with value 1 is the bit corresponding to the 15 value.

We create the new training set repeating each row 10 times because, after various tests, we noticed that it was the best choice. We "choose a small random subset of features" for each row based on a random number between 1 and the natural logarithm of the number of features, i.e. `maxSubsetFeaturesToDelete = int(np.round(np.log(len(cvFeatures))))` where `len(lbFeatures)` is the features number of the dataset.

After processing and creating the new training set, we trained the implemented MLP.

Results obtained by Lorenzo using **Letter Recognition DataSet** with 20000 samples and 16 features, where 18000 samples are for the training set and 2000 samples are for the test set, is shown on this table:

# of repetitions for each row	**Learning rate**	**Tolerance for the optimization**	**# of layers and nodes**	**Accuracy of predictions**	**# of rows predicted correctly**
10	**0.0003**	**0.00003**	**1 layer with # of bits in input * 20**	**28.8%**	**705 on 2000, 35.25%**
10	**0.0003**	**0.00003**	**1 layer with # of bits in input * 40**	**28.35%**	**719 on 2000, 35.95%**
20	0.0003	0.00003	1 layer with # of bits in input * 20	28.75%	652 on 2000, 32.6%
2	0. 001	0.005	1 layer with # of bits in input * 20	23.8%	554 on 2000, 27.70%

The first 2 rows of this table are the best result we obtained.

Results obtained by Manuel using **Connect-4 DataSet** with 67557 samples and 41 features, where 60000 samples are for the training set and 7557 samples are for the test set, is shown on this table:

# of repetitions for each row	**Learning rate**	**Tolerance for the optimization**	**# of layers and nodes**	**Accuracy of predictions**	**# of rows predicted correctly**
2	0.001	0.005	1 layer with # of bits in input * 10	4.64%	86 on 7557, 1.14%
3	0.001	0.001	1 layer with # of bits in input * 20	0.50%	14 on 7557, 0.19%
1	**0.001**	**0.001**	**1 layer with # of bits in input * 14**	**5.81%**	**125 on 7557, 1.65%**
1	0.001	0.001	3 layers: 1) with # of bits in input * 6 2) with # of bits in input * 12 3) with # of bits in input * 6	1.07%	48 on 7557, 0.64%

The third row of this table is the best result we obtained.