CSCI 2410 Introduction to Data Analytics Using Python
Homework Assignment #6

HW Programming #6: - Data Analytics with Artificial Neural Networks

Tasks: Experiment with the Artificial Neural Networks – MLP technique on 'digits' dataset loaded from sklearn datasets.

Assignment Instructions:

1. **[80%]** Run **MLP Neural Network technique** on 'digits' dataset loaded from sklearn datasets
   The dataset 'digits' is discussed in the lecture notes.
   The dataset contains 1,797 samples, each sample has 64 features (i.e., 1,797 rows and 64 columns in the Excel data file), with an additional field called 'Target' (i.e., the 'Label' of the sample) indicate what the digit is for that sample.
   Try different Neural Network parameters, i.e.,
   - **(1)** Number of layers: from 1 layer to 2 layers, 3 layers, 4 layers, Note: it will take longer time to run with more layers.
   - **(2)** Number of neurons at each layer – may range from 10 to 100. (It is suggested to have a relatively larger number of neurons on the first and last hidden layers so as to get a good accuracy score.)
   - **(3)** Run your script with 10-20 runs – each run with different number of iterations. Record your runs in a table (or Excel sheet) such as an example below:

Run #	# of layers	# of neurons on each layer	# iterations	Accuracy	alpha	Output function (activation)	Other parameters (learning rate)
1	1	(50)	200	0.98	0.0001	relu	constant
2	1	(100)	200	0.97	0.0001	tanh	invscaling
3	2	(30, 20)	300	0.95	0.0001	identity	adaptive
4	3	(50, 30, 20)	500	0.96	0.0001	logistic	constant
5	3	(10, 50, 10)	1000	0.94	0.0001	relu	invscaling
6	3	(70, 40, 20)	800	0.96	0.0001	tanh	adaptive
7	1	(50)	700	0.97	0.0001	identity	constant
8	2	(30, 30)	600	0.96	0.0001	logistic	invscaling
9	2	(80, 20)	400	0.96	0.0001	relu	adaptive
10	3	(30, 20, 10)	500	0.95	0.0001	tanh	constant
11	4	(20, 50, 20, 50)	250	0.97	0.0001	identity	invscaling
12	1	(100)	350	0.96	0.0001	logistic	adaptive
13	4	(10, 10, 5, 5)	500	0.89	0.0001	relu	constant
14	2	(10, 10)	950	0.97	0.0001	tanh	invscaling
15	2	(20, 20)	250	0.93	0.0001	identity	adaptive
16	3	(20, 10, 25)	550	0.91	0.0001	logistic	constant
17	3	(10, 50, 10)	250	0.94	0.0001	relu	invscaling
18	4	(10, 10, 10, 10)	450	0.91	0.0001	tanh	adaptive
19	4	(40, 60, 40, 20)	555	0.97	0.0001	identity	constant
20	3	(30, 10, 20)	666	0.94	0.0001	logistic	invscaling

The "Other parameters" could be any of those that can be set in the MLPClassifier() function.

2. [20%] Discussion Problem

Summarize and compare the performances (e.g., the classification accuracy) of the runs, and discuss how they are related to the different parameters (# of layers, # of neurons on each layer, number of run iterations, activation function, etc.) of the neural networks.

1. Number of Layers
    - The performance tends to improve with an increase in the number of layers up to a certain point.
2. Number of Neurons on Each Layers
    - Larger numbers of neurons, especially on the first and last layers, generally lead to better performance, as they allow the network to learn more intricate representations of the data.
3. Number of Iterations
    - Performance tends to improve with more iterations, as the model has more opportunities to learn from the data. However, there's a point of diminishing returns, and too many iterations might lead to overfitting.
4. Activation Function
    - The choice of activation function plays a significant role in model performance. The ReLU activation function tends to work well in hidden layers, capturing non-linear relationships.
5. Learning Rate
    - The learning rate impact the convergence of the model during training. A well-tuned learning rate can lead to faster convergence and better performance. Learning rate schedules, such as 'invscaling' or 'adaptive,' can adapt the learning rate during training, potentially improving convergence on certain datasets.
6. Conclusion
    - The optimal configuration depends on the specific characteristics of the dataset. Experimenting with different combinations of parameters and monitoring performance is crucial

Python libraries needed: sklearn-datasets, sklearn.model_selection-train_test_split sklearn.preprocessing-StandardScaler, sklearn.neural_network import MLPClassifier, sklearn.metrics-accuracy_score, sklearn.metrics-confusion_matrix, matplotlib.pyplot

Please look at the “2410 HM Assignment #6 Supplemental Guidance” for more information for this assignment.

Requirements for the Submission of Programming/Homework Assignments

1. Well-documented program list (the .py files)
   20% of total points if no .py file submitted.
   Done

2. Three annotated program test and run examples (screenshots) that **show different and representative test cases** with **input, output, and the parameter settings of the program runs clearly marked/annotated**. You can do the annotations by
   (1) Pasting the screenshots into a WORD document,
   Done
   (2) Editing on the WORD document pages for the required marks and annotations,
   Done
   (3) Converting the document to pdf for submission (it is ok to submit the WORD file directly without converting to pdf).
   Done
   20% of total points will be taken off if run examples are not representative.
   20% of total points will be taken off if run examples are not clearly marked/annotated.

3. A discussion page
   (a) Hardware and software used by your program,
   I completed this assignment using my personal computer with PyCharm Professional Version: 2023.2.1.
   (b) Features of your program, e.g., data structures, algorithms, programming styles, etc.
   The program utilizes scikit-learn libraries for machine learning tasks, applying the Multi-Layer Perceptron algorithm with specified configurations, featuring modularized digit image display using matplotlib, employing standardization with StandardScaler, and dynamically presenting key information.
   (c) Problems you encountered during your work, and
   None
   (d) Assigned discussion problems, if there is any.
   Assigned discussion problems were answered
   (e) Fill in the following table and submit it along with your above submissions.

Total (approximate) time spent on the assignment	8 hours	Total (approximate) time for the correction part	1 hour
Problems and difficulties encountered	None		
Reflections (good and bad) on the assignment	Good: A snippet of lines of code was provided Bad: None		

Any comments and suggestions	None

20% of total points will be taken off if no discussion page is submitted.