Machine Learning Lab (UE22CS352A)

Artificial Neural Network

Teaching Assistant:
arya.tapikar@gmail.com

An Artificial Neural Network (ANN) is a computational model inspired by neural connections in the brain. A Multilayer Perceptron (MLP) Classifier is a type of artificial neural network specifically designed for classification tasks. It's a feedforward neural network with an input layer, one or more hidden layers, and an output layer. Each layer consists of interconnected nodes, or neurons, which apply weighted transformations to inputs and pass the results through activation functions. In the context of classification, the output layer typically employs an activation function like softmax to produce probability distributions over classes. During training, the MLP Classifier adjusts its internal weights using optimization techniques like backpropagation and gradient descent to minimize the classification error. This allows the MLP Classifier to learn complex decision boundaries and relationships within the input data, making it effective for tasks such as image recognition, natural language processing, and more.

Scikit-learn's `MLPClassifier` is a user-friendly yet powerful tool for classification tasks. Its simplicity in implementation, efficient handling of large datasets, customization options for architecture and hyperparameters, hyperparameter tuning tools, and strong documentation make it an excellent choice for various classification challenges.

Task:

Complete code for functions whose skeleton has been provided.

You are provided with the following files:

1. ANN.py
2. Test.py

ANN.py

This file contains the following functions:

Function name	Input	Output
Load_and_preprocess_data	filepath	tuple of X (features), y (target)
split_and_standardize	1. x: list/ndarray (features) 2. y: list/ndarray (target)	split: tuple of X_train, X_test, y_train, y_test
create_model	1. X_train: list/ndarray 2. y_train: list/ndarray	models: model1,model2 - tuple
predict_and_evaluate	1. model: MLPClassifier aftertraining 2. X_train: list/ndarray 3. y_train: list/ndarray	metrics: tuple - accuracy,precision,recall,fscore,confusion matrix

Note: The function create_model returns a tuple of two MLPClassifier objects (Two models). Create and train 2 MLP classifier (of 3 hidden layers each) with different parameters. The first model must have high accuracy (>=80%), the second model's accuracy must be in the range 50% to 95%

Test.py

1. This will help you check your code.
2. Rename ANN.py file to CAMPUS_SECTION_SRN_Lab4.py
3. Run the command `python3 Test.py --ID CAMPUS_SECTION_SRN_Lab4 --filepath path_to_dataset.csv`

The sample dataset used is a modified version of a classic diabetes disease dataset. This dataset contains various health-related attributes like glucose levels, blood pressure, BMI, and others, with an `"Outcome"` column indicating whether a patient has diabetes (`1.0`) or not (`0.0`). Additionally, it includes a `"GarbageValues"` column, which likely contains irrelevant or erroneous data that should be removed during preprocessing.

Note

- The dataset contains discrepancies introduced for testing, such as the "GarbageValues" column with erroneous data.
- For analysis or model training, the "GarbageValues" column should be removed, and rows with missing or invalid values along with duplicate data should be handled appropriately.

Important Points

1. **Do not make changes to the function definitions** that are provided to you. Use the skeleton as it has been given.
2. **Do not make changes to the test file provided to you**. Run as is.
3. Do not hardcode values. The functions must be designed to be independent of the schema of the dataset.
4. You may write additional helper functions.
5. You **can use built-in modules**.
6. You **must use sklearn**

Submission Guidelines

You are to submit two files:

1. The python solution: CAMPUS_SECTION_SRN_Lab4.py
2. Screenshot of Test cases (Single screenshot of all testcases in terminal): CAMPUS_SECTION_SRN_Lab4.png (or jpg)

The google form link for submission will be provided

- Remove all print statements.
- Resubmitting from different mail will lead to zero marks.
- Failing some hidden cases will lead to partial marks.
- Test cases provided to you are for reference only, hidden test cases will be similar