COSC 425: Intro to Machine Learning
Lab 5: Multi-Layer Perceptrons

Due: November 11, 2022, 11:59 PM

Introduction

In this lab, you will be practicing using sklearn's MLPClassifier and applying it to the GTZAN music genre classification dataset: `https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification`

You will create a Jupyter notebook that you will submit where you will put the code that you wrote for this lab. You will also submit a separate written report in PDF form in which you include the answers to the questions and/or plots required by the question.

Note: For all plots, you should include axes labels and titles.

Questions

Read in the features_3_sec.csv dataset using pandas and create the y label vector that we will use throughout, which will be the label column. Remove the filename and label columns from the dataframe (using drop) and create the X features matrix from the remainder of the columns. Create a train-test split on the data; use a fixed random state (noted in your write-up) and a test size=0.2. In all cases, unless otherwise specified, leave the MLPClassifier parameters to their defaults, except use activation="tanh", solver="sgd", and fix a random state value of your choice.

Question 1 (5 points): For each of the features of the data (length, chroma_stft_mean, etc.), calculate and print the mean and standard deviation of that feature value in X_train. Describe (in 1-3 sentences) why these values might cause an issue when using the raw values for neural network classification.

Question 2 (5 points): Using sklearn's preprocessing.StandardScaler(), fit to the X_train data, and transform both the X_train and X_test data based on the pre-processing fit. Again, for each of the features of the data (length, chroma_stft_mean, etc.), calculate and print the mean and standard deviation of that feature value in the updated, pre-processed X_train. Describe briefly (in 1-3 sentences) what happened and why it might be a good idea to do this transformation for a multi-layer perceptron.

Question 3 (30 points): Use K-fold cross-validation with three folds to find the best values to use for the number of neurons in a single hidden layer across 50, 100, 200, and 500 neurons and for the initial learning rate (learning_rate_init) across 0.0001, 0.001, 0.01, 0.1, and 1. Create a heatmap showing the average validation accuracy across all folds for each parameter combination. Note in your report which combination performs best.

Question 4 (25 points): Use K-fold cross-validation with three folds to find the structure of the network to use. Set the initial learning rate to be the best value you found in Question 3. Try the following combinations of network structure: (100), (100,100), (100,100,100), (200), (200,100), (200,100,100), (500), (500,200), (500,200,100). Note the structure of the network that gives the best results in the report.

Question 5 (20 points): Using the best structure and best initial learning rates found above, now train MLPs on the entire training set for 1000 iterations (max_iter=1000). Generate 10 different random integers to use as initial random states for each MLP, but otherwise, use the same parameters. Show the training and testing accuracy for each of the 10 classifiers. Plot the 10 resulting loss curves (note that you can access the loss curve from the classifier in clf.loss_curve_).

Question 6 (15 points): Show the confusion matrix for the classifier. Print the matrix itself. Plot the confusion matrix with a heatmap (I recommend using seaborn's heatmap for this). Which classes are most often confused for each other? Include a brief discussion of this in the report (1-3 sentences).

Submission Checklist

In this lab, you will submit on Canvas:

- Jupyter notebook
- PDF of your report

Late Penalty: 20 points off per day late.