Name: Sarah Huang

Date: 11/10/2023

# Lab 5: Multi-Layer Perceptrons

The fixed random state I use in my calculations is **42**.

**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.

**length**

Mean: 66149.0

Standard Deviation: 0.0

**chroma\_stft\_mean**

Mean: 0.3793348486780434

Standard Deviation: 0.08994476733531448

**chroma\_stft\_var**

Mean: 0.08491067684514704

Standard Deviation: 0.009636830553150478

**rms\_mean**

Mean: 0.13045734546834067

Standard Deviation: 0.06831589241988416

**rms\_var**

Mean: 0.0026741077189679236

Standard Deviation: 0.003602569391982133

**spectral\_centroid\_mean**

Mean: 2197.217605496407

Standard Deviation: 750.5751506552442

**spectral\_centroid\_var**

Mean: 416394.5709013398

Standard Deviation: 435125.0983493911

**spectral\_bandwidth\_mean**

Mean: 2241.4160274982455

Standard Deviation: 542.0331439300644

**spectral\_bandwidth\_var**

Mean: 118876.02725554115

Standard Deviation: 101086.7718012885

**rolloff\_mean**

Mean: 4562.063904866029

Standard Deviation: 1638.4681035407077

**rolloff\_var**

Mean: 1635723.1968233867

Standard Deviation: 1494492.381244041

**zero\_crossing\_rate\_mean**

Mean: 0.10242616696474113

Standard Deviation: 0.04557790251849784

**zero\_crossing\_rate\_var**

Mean: 0.002604532182368757

Standard Deviation: 0.003562302426243175

**harmony\_mean**

Mean: -0.0003549154551852361

Standard Deviation: 0.0016503630311444056

**harmony\_var**

Mean: 0.012541495779202192

Standard Deviation: 0.012604115062498552

**perceptr\_mean**

Mean: -0.00038656910958775333

Standard Deviation: 0.0010905758930807269

**perceptr\_var**

Mean: 0.005552603638752261

Standard Deviation: 0.006582949497354747

**tempo**

Mean: 124.66841439164905

Standard Deviation: 32.89022643143318

**mfcc1\_mean**

Mean: -145.76008223406646

Standard Deviation: 106.3327661142152

**mfcc1\_var**

Mean: 2825.53030584405

Standard Deviation: 2650.2581081581407

**mfcc2\_mean**

Mean: 101.07427029043913

Standard Deviation: 34.60374342001305

**mfcc2\_var**

Mean: 590.8061189126444

Standard Deviation: 462.4406057655496

**mfcc3\_mean**

Mean: -9.97095077588437

Standard Deviation: 23.85372504938617

**mfcc3\_var**

Mean: 375.4690324021233

Standard Deviation: 298.64767055276855

**mfcc4\_mean**

Mean: 37.1440390884608

Standard Deviation: 17.774523889346625

**mfcc4\_var**

Mean: 183.94436323210164

Standard Deviation: 133.22715989900016

**mfcc5\_mean**

Mean: -2.0309466108609358

Standard Deviation: 13.625243467581992

**mfcc5\_var**

Mean: 144.28271135362658

Standard Deviation: 110.55480005788691

**mfcc6\_mean**

Mean: 15.348786673807139

Standard Deviation: 12.612486638829804

**mfcc6\_var**

Mean: 107.941944606192

Standard Deviation: 76.65103562736212

**mfcc7\_mean**

Mean: -5.778828046345416

Standard Deviation: 11.114671467400743

**mfcc7\_var**

Mean: 98.07257676303566

Standard Deviation: 65.14385812417788

**mfcc8\_mean**

Mean: 10.769641479648843

Standard Deviation: 11.094700452783309

**mfcc8\_var**

Mean: 74.48950701468699

Standard Deviation: 45.531131330988444

**mfcc9\_mean**

Mean: -7.614995407015335

Standard Deviation: 9.380115663000206

**mfcc9\_var**

Mean: 74.08010648320744

Standard Deviation: 44.48749398712771

**mfcc10\_mean**

Mean: 8.304197901847498

Standard Deviation: 8.871942867057806

**mfcc10\_var**

Mean: 68.73229113617936

Standard Deviation: 41.65544145494373

**mfcc11\_mean**

Mean: -6.494609810939586

Standard Deviation: 7.836534932517653

**mfcc11\_var**

Mean: 63.74561351740563

Standard Deviation: 40.30089465806018

**mfcc12\_mean**

Mean: 4.982660296956555

Standard Deviation: 7.565662442692437

**mfcc12\_var**

Mean: 57.684814189408755

Standard Deviation: 37.0183144829037

**mfcc13\_mean**

Mean: -5.174321344729334

Standard Deviation: 7.174499213727445

**mfcc13\_var**

Mean: 57.13016234247296

Standard Deviation: 36.14702955390238

**mfcc14\_mean**

Mean: 2.1705309923065226

Standard Deviation: 6.106530936457899

**mfcc14\_var**

Mean: 53.87679445296079

Standard Deviation: 37.627022457131766

**mfcc15\_mean**

Mean: -4.160332547940197

Standard Deviation: 5.962368329001718

**mfcc15\_var**

Mean: 52.46316307571318

Standard Deviation: 36.527930370988614

**mfcc16\_mean**

Mean: 1.444757134132186

Standard Deviation: 5.754681741245769

**mfcc16\_var**

Mean: 49.78360258100985

Standard Deviation: 34.48603624593198

**mfcc17\_mean**

Mean: -4.212607759286245

Standard Deviation: 5.704707604977488

**mfcc17\_var**

Mean: 51.792301278781366

Standard Deviation: 36.24649760989759

**mfcc18\_mean**

Mean: 0.7333429515319776

Standard Deviation: 5.228567784585092

**mfcc18\_var**

Mean: 52.22550680287727

Standard Deviation: 37.53951344601074

**mfcc19\_mean**

Mean: -2.4888398104986655

Standard Deviation: 5.134577891949479

**mfcc19\_var**

Mean: 55.20132899081504

Standard Deviation: 42.40123814967258

**mfcc20\_mean**

Mean: -0.8937203961417457

Standard Deviation: 5.260200946027244

**mfcc20\_var**

Mean: 57.386735770653736

Standard Deviation: 45.98457744940269

They are all drastically different values like the length has a mean in the thousands while as the chroma\_stft\_mean has a mean less than one. If there was a scale, they wouldn’t appear so extreme and make certain features overpower others in the learning process.

**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.

**length**

Mean: 0.0

Standard Deviation: 0.0

**chroma\_stft\_mean**

Mean: -1.0846623343685213e-16

Standard Deviation: 1.0

**chroma\_stft\_var**

Mean: -4.106824990189994e-15

Standard Deviation: 0.9999999999999982

**rms\_mean**

Mean: -5.778938666717531e-16

Standard Deviation: 0.9999999999999974

**rms\_var**

Mean: -1.2438054145750503e-14

Standard Deviation: 0.9999999999999988

**spectral\_centroid\_mean**

Mean: 1.131782911189449e-15

Standard Deviation: 0.9999999999999981

**spectral\_centroid\_var**

Mean: -1.4636273507821133e-15

Standard Deviation: 1.0000000000000013

**spectral\_bandwidth\_mean**

Mean: 3.24865259818244e-15

Standard Deviation: 0.9999999999999992

**spectral\_bandwidth\_var**

Mean: -5.379747364970428e-15

Standard Deviation: 0.9999999999999996

**rolloff\_mean**

Mean: 5.6926991204603624e-15

Standard Deviation: 1.0000000000000009

**rolloff\_var**

Mean: 4.476454797988119e-16

Standard Deviation: 0.9999999999999997

**zero\_crossing\_rate\_mean**

Mean: -8.340342097304179e-15

Standard Deviation: 0.999999999999998

**zero\_crossing\_rate\_var**

Mean: 4.662269902810267e-15

Standard Deviation: 0.9999999999999984

**harmony\_mean**

Mean: 4.12527314054913e-15

Standard Deviation: 0.9999999999999987

**harmony\_var**

Mean: -3.685184734391411e-15

Standard Deviation: 1.0000000000000018

**perceptr\_mean**

Mean: 8.912012491565621e-15

Standard Deviation: 0.9999999999999996

**perceptr\_var**

Mean: -6.507084938723941e-15

Standard Deviation: 0.9999999999999989

**tempo**

Mean: -1.2258995953830977e-13

Standard Deviation: 1.00000000000002

**mfcc1\_mean**

Mean: -5.6900319179988e-17

Standard Deviation: 0.999999999999998

**mfcc1\_var**

Mean: 7.912700635967082e-17

Standard Deviation: 1.0000000000000002

**mfcc2\_mean**

Mean: 8.268327630842006e-17

Standard Deviation: 1.0000000000000004

**mfcc2\_var**

Mean: -5.334404923123875e-17

Standard Deviation: 0.9999999999999998

**mfcc3\_mean**

Mean: -6.668006153904844e-17

Standard Deviation: 0.9999999999999996

**mfcc3\_var**

Mean: 7.82379388724835e-17

Standard Deviation: 0.9999999999999987

**mfcc4\_mean**

Mean: -5.601125169280069e-17

Standard Deviation: 1.0000000000000004

**mfcc4\_var**

Mean: 5.756711979537848e-17

Standard Deviation: 0.9999999999999992

**mfcc5\_mean**

Mean: 9.55747548726361e-18

Standard Deviation: 0.9999999999999991

**mfcc5\_var**

Mean: -5.867845415436262e-17

Standard Deviation: 1.000000000000001

**mfcc6\_mean**

Mean: 1.778134974374625e-17

Standard Deviation: 1.0000000000000022

**mfcc6\_var**

Mean: 4.623150933374025e-17

Standard Deviation: 0.9999999999999982

**mfcc7\_mean**

Mean: 8.890674871873125e-18

Standard Deviation: 0.999999999999999

**mfcc7\_var**

Mean: 7.1125398974985e-18

Standard Deviation: 1.0000000000000007

**mfcc8\_mean**

Mean: -1.8670417230933565e-17

Standard Deviation: 0.9999999999999991

**mfcc8\_var**

Mean: 1.0313182851372826e-16

Standard Deviation: 0.9999999999999991

**mfcc9\_mean**

Mean: 4.712057682092756e-17

Standard Deviation: 1.000000000000001

**mfcc9\_var**

Mean: 1.391390617448144e-16

Standard Deviation: 0.9999999999999989

**mfcc10\_mean**

Mean: -3.289549702593056e-17

Standard Deviation: 0.9999999999999989

**mfcc10\_var**

Mean: 1.4758520287309388e-16

Standard Deviation: 1.0000000000000013

**mfcc11\_mean**

Mean: -1.5114147282184315e-17

Standard Deviation: 0.9999999999999987

**mfcc11\_var**

Mean: 3.911896943624175e-17

Standard Deviation: 1.0000000000000016

**mfcc12\_mean**

Mean: -1.1913504328309988e-16

Standard Deviation: 1.0000000000000013

**mfcc12\_var**

Mean: -2.2226687179682815e-17

Standard Deviation: 0.9999999999999993

**mfcc13\_mean**

Mean: -3.022829456436863e-17

Standard Deviation: 0.999999999999999

**mfcc13\_var**

Mean: -5.601125169280069e-17

Standard Deviation: 0.999999999999999

**mfcc14\_mean**

Mean: -6.668006153904844e-19

Standard Deviation: 1.0000000000000013

**mfcc14\_var**

Mean: -6.045658912873726e-17

Standard Deviation: 1.0000000000000007

**mfcc15\_mean**

Mean: 4.223070564139735e-18

Standard Deviation: 0.9999999999999996

**mfcc15\_var**

Mean: 6.467965969287698e-17

Standard Deviation: 0.9999999999999978

**mfcc16\_mean**

Mean: -6.668006153904844e-18

Standard Deviation: 1.000000000000001

**mfcc16\_var**

Mean: -4.356430687217831e-17

Standard Deviation: 1.0000000000000022

**mfcc17\_mean**

Mean: 4.689830994913074e-17

Standard Deviation: 1.0

**mfcc17\_var**

Mean: 4.445337435936563e-19

Standard Deviation: 0.9999999999999996

**mfcc18\_mean**

Mean: 3.645176697467981e-17

Standard Deviation: 1.0000000000000002

**mfcc18\_var**

Mean: -6.045658912873726e-17

Standard Deviation: 1.0000000000000009

**mfcc19\_mean**

Mean: 2.933922707718131e-17

Standard Deviation: 1.000000000000001

**mfcc19\_var**

Mean: -2.756109210280669e-17

Standard Deviation: 0.9999999999999981

**mfcc20\_mean**

Mean: -3.911896943624175e-17

Standard Deviation: 1.0

**mfcc20\_var**

Mean: 1.778134974374625e-18

Standard Deviation: 1.0000000000000009

By using StandardScaler(), the feature values are all on the same scale and similar in values. Now the neural network will learn more effectively since the values aren’t all drastically different.

**Question 3 (30 points):** Use K-fold cross-validation with three folds to find the best values

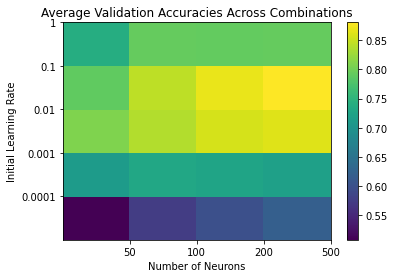
to use for the number of neurons in a single hidden layer across 50, 100, 200, and 500

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.

The combination that works the best is an initial learning rate of 0.1 and 500 neurons.



**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.

**100 -** 0.8435935935935936

**(100, 100) -** 0.8606106106106107

**(100, 100, 100) -** 0.8638638638638638

**200 -** 0.8683683683683684

**(200, 100) -** 0.8770020020020021

**(200, 100, 100) -** 0.8788788788788789

**500 -** 0.8805055055055057

**(500, 200) -** 0.8885135135135135

**(500, 200, 100) -** 0.8906406406406407

The structure that worked the best was (500, 200, 100).

**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 ).

**218**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9224224224224224

**132**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9254254254254254

**508**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.923923923923924

**604**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9269269269269269

**68**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9219219219219219

**132**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9254254254254254

**94**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9219219219219219

**709**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9224224224224224

**809**

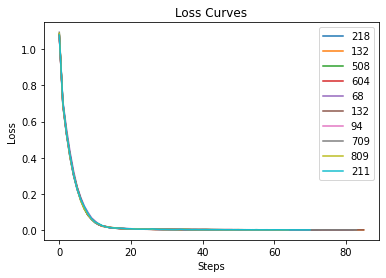
Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9269269269269269

**211**

Training Accuracy: 0.9992492492492493

Testing Accuracy: 0.9259259259259259



**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).

In the training data, the classes that are most often confused for each other are metal and rock because there are 6 instances of the model predicting it to be metal when it’s actually rock. In the testing data, the classes that are most often confused for each other are blues and country because there are 11 instances of the model predicting it to be country when it’s actually blues. Overall, the model is correctly guessing the genre for both training and testing data because the outputs are very high.

Training Confusion Matrix

[[792 0 0 0 0 0 0 0 0 0]

[ 0 795 0 0 0 0 0 0 0 0]

[ 0 0 811 0 0 0 0 0 0 0]

[ 0 0 0 800 0 0 0 0 0 0]

[ 0 0 0 0 780 0 0 0 0 0]

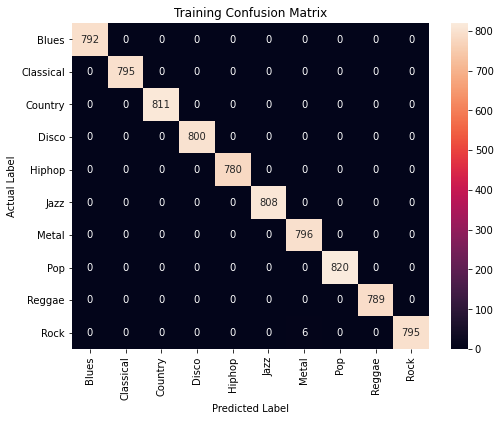
[ 0 0 0 0 0 808 0 0 0 0]

[ 0 0 0 0 0 0 796 0 0 0]

[ 0 0 0 0 0 0 0 820 0 0]

[ 0 0 0 0 0 0 0 0 789 0]

[ 0 0 0 0 0 0 6 0 0 795]]



Testing Confusion Matrix

[[185 0 11 2 0 4 1 0 2 3]

[ 0 196 0 0 0 7 0 0 0 0]

[ 5 1 167 4 0 3 0 1 1 4]

[ 3 3 1 177 5 0 1 2 5 2]

[ 1 1 3 1 205 0 2 4 1 0]

[ 4 6 2 1 0 177 0 0 1 1]

[ 1 0 1 0 0 0 199 0 0 3]

[ 1 0 0 2 3 1 0 169 4 0]

[ 0 1 1 6 4 2 0 1 192 4]

[ 2 1 3 4 3 2 10 1 3 168]]

