Group Members: Connor Smith cms6788 Abhishek Sridhar as83788 Tom Li hl26868

In [3]:
```
#1 import data/libraries
import random
import sklearn
import sklearn.datasets
import matplotlib.pyplot as plt
from PIL import Image
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LogisticRegressionCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import RandomizedSearchCV
import xgboost as xgb
from xgboost import XGBClassifier
# import torch
```

In [11]:
```
dataset = sklearn.datasets.fetch_openml("CIFAR_10_small", cache=True)
```

In [4]:
```
dataset_num = sklearn.datasets.fetch_openml('mnist_784', cache=True)
```

In [12]:
```
# reduce CIFAR size
print(dataset.data.shape, dataset.target.shape)
data = dataset.data.join(dataset.target)
data = data.sample(frac=0.5, random_state=0, replace=False)
dataset.data, dataset.target = data.iloc[:, :-1], data.iloc[:, [-1]].squeeze()
print(dataset.data.shape, dataset.target.shape)
```

```
(20000, 3072) (20000,)
(10000, 3072) (10000,)
```

In [ ]:
```
print(type(dataset_num.target))
```

In [5]:
```
# reduce MNIST size
print('before: ',dataset_num.data.shape, dataset_num.target.shape)
data_num = dataset_num.data.join(dataset_num.target)
data_num = data_num.sample(frac=0.5, random_state=0, replace=False)
dataset_num.data, dataset_num.target = data_num.iloc[:, :-1], data_num.iloc[:, [
print('after: ', dataset_num.data.shape, dataset_num.target.shape)
```

```
before:  (70000, 784) (70000,)
after:  (35000, 784) (35000,)
```

In [26]:
```
fig = plt.figure()
for i in range(5):
    fig.add_subplot(1, 5, i+1)
    rand = random.randrange(0,10000,1)
    classNames = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog'
```

```
    classifications = dataset["target"].iloc[rand]
    im = np.uint8(np.transpose(dataset["data"].iloc[rand].to_numpy().reshape(3,
    plt.imshow(im)
    plt.title(classNames[int(classifications)])
```

In [14]:

```
#create train test splits for data
train_img, test_img, train_lbl, test_lbl = train_test_split(dataset.data, datase
train_img_num, test_img_num, train_lbl_num, test_lbl_num = train_test_split(data
```

In [27]:

```
%%time
#Train Logistic Regression Model
clf_l1 = LogisticRegression(fit_intercept=True,
                            multi_class= 'multinomial',
                            penalty= 'l2',
                            solver= 'saga',
                            max_iter= 100,
                            C=.001,
                            verbose=1, n_jobs=4
                            )

clf_l1.fit(train_img, train_lbl)
```

```
[Parallel(n_jobs=4)]: Using backend ThreadingBackend with 4 concurrent workers.
Epoch 1, change: 1.00000000
Epoch 2, change: 0.32671294
Epoch 3, change: 0.16623850
Epoch 4, change: 0.15291279
Epoch 5, change: 0.11874959
Epoch 6, change: 0.09164642
Epoch 7, change: 0.08095785
Epoch 8, change: 0.07029986
Epoch 9, change: 0.06454135
Epoch 10, change: 0.05694027
Epoch 11, change: 0.05139672
Epoch 12, change: 0.04718630
Epoch 13, change: 0.04367321
Epoch 14, change: 0.04014699
Epoch 15, change: 0.03805809
Epoch 16, change: 0.03493958
Epoch 17, change: 0.03313849
Epoch 18, change: 0.03150472
Epoch 19, change: 0.02968605
Epoch 20, change: 0.02819296
Epoch 21, change: 0.02696805
Epoch 22, change: 0.02546941
Epoch 23, change: 0.02431766
Epoch 24, change: 0.02330152
Epoch 25, change: 0.02255569
Epoch 26, change: 0.02134452
Epoch 27, change: 0.02070704
```

```
Epoch 28, change: 0.01990694
Epoch 29, change: 0.01925419
Epoch 30, change: 0.01861840
Epoch 31, change: 0.01791764
Epoch 32, change: 0.01748502
Epoch 33, change: 0.01678554
Epoch 34, change: 0.01643839
Epoch 35, change: 0.01573706
Epoch 36, change: 0.01540116
Epoch 37, change: 0.01478301
Epoch 38, change: 0.01449233
Epoch 39, change: 0.01396864
Epoch 40, change: 0.01371648
Epoch 41, change: 0.01333702
Epoch 42, change: 0.01291432
Epoch 43, change: 0.01270065
Epoch 44, change: 0.01224056
Epoch 45, change: 0.01196833
Epoch 46, change: 0.01165515
Epoch 47, change: 0.01141222
Epoch 48, change: 0.01110437
Epoch 49, change: 0.01089632
Epoch 50, change: 0.01064359
Epoch 51, change: 0.01042824
Epoch 52, change: 0.01023076
Epoch 53, change: 0.00992123
Epoch 54, change: 0.00982935
Epoch 55, change: 0.00963599
Epoch 56, change: 0.00945876
Epoch 57, change: 0.00931693
Epoch 58, change: 0.00917834
Epoch 59, change: 0.00909846
Epoch 60, change: 0.00895175
Epoch 61, change: 0.00879017
Epoch 62, change: 0.00874974
Epoch 63, change: 0.00860333
Epoch 64, change: 0.00841560
Epoch 65, change: 0.00836864
Epoch 66, change: 0.00828102
Epoch 67, change: 0.00813513
Epoch 68, change: 0.00805490
Epoch 69, change: 0.00790803
Epoch 70, change: 0.00787164
Epoch 71, change: 0.00780347
Epoch 72, change: 0.00767606
Epoch 73, change: 0.00762322
Epoch 74, change: 0.00753291
Epoch 75, change: 0.00743082
Epoch 76, change: 0.00735054
Epoch 77, change: 0.00724438
Epoch 78, change: 0.00714460
Epoch 79, change: 0.00712964
Epoch 80, change: 0.00705699
Epoch 81, change: 0.00697801
Epoch 82, change: 0.00688890
Epoch 83, change: 0.00681770
Epoch 84, change: 0.00675845
Epoch 85, change: 0.00669039
Epoch 86, change: 0.00662220
Epoch 87, change: 0.00657566
Epoch 88, change: 0.00649374
```

```
Epoch 89, change: 0.00640389
Epoch 90, change: 0.00638387
Epoch 91, change: 0.00635279
Epoch 92, change: 0.00625368
Epoch 93, change: 0.00620312
Epoch 94, change: 0.00618348
Epoch 95, change: 0.00608324
Epoch 96, change: 0.00603021
Epoch 97, change: 0.00599505
Epoch 98, change: 0.00593854
Epoch 99, change: 0.00589878
max_iter reached after 80 secondsEpoch 100, change: 0.00581582

CPU times: user 1min 19s, sys: 182 ms, total: 1min 20s
Wall time: 1min 20s
```

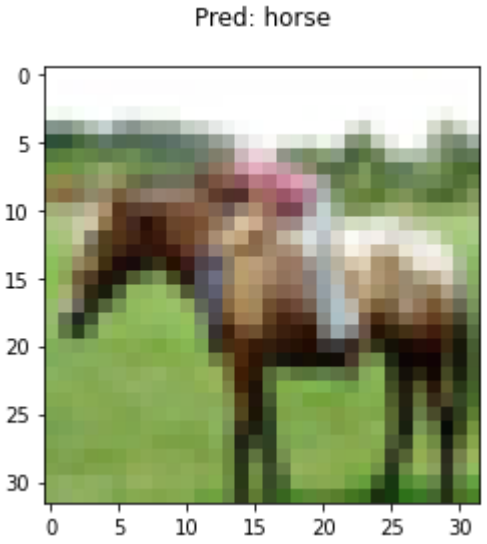
```
/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
[Parallel(n_jobs=4)]: Done   1 out of   1 | elapsed:  1.3min finished
```

Out[27]:
```
LogisticRegression(C=0.001, multi_class='multinomial', n_jobs=4, solver='saga',
                   verbose=1)
```

In [28]:
```
#Prediction test
pred0 = clf_l1.predict(np.array(test_img.iloc[0]).reshape(1,-1))

print(pred0)
fig = plt.figure()

img = test_img.iloc[0].tolist()
img = np.array(img)
img = img.reshape(3, 32, 32)
img = np.transpose(img, axes = [1, 2, 0])
img = np.uint8(img)
plt.imshow(img)
plt.title('Pred: ' + classNames[int(pred0)] + '\n')
```

```
['7']
```

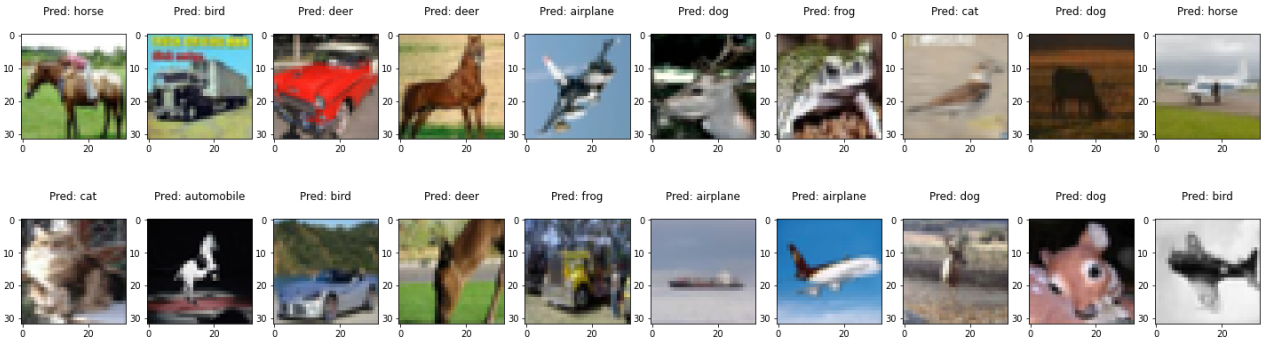
Out[28]:
```
Text(0.5, 1.0, 'Pred: horse\n')
```



In [29]:
```
from sklearn.metrics import log_loss
```

```
label_pred = clf_l1.predict(test_img)
print("accuracy: ", accuracy_score(test_lbl.to_numpy(), label_pred))
print("training loss: ", log_loss(train_lbl.to_numpy(), clf_l1.predict_proba(tra
print("test loss: ", log_loss(test_lbl.to_numpy(), clf_l1.predict_proba(test_img
test_fig = plt.figure(figsize=(25,7))
for i in range(20):
    test_fig.add_subplot(2, 10, i+1)
    img = np.transpose(test_img.iloc[i].to_numpy().reshape(3, 32, 32), axes = [1
    img = np.uint8(img)
    plt.imshow(img, cmap=plt.cm.gray)
    plt.title('Pred: ' + classNames[int(label_pred[i])] + '\n')
```

```
accuracy:  0.3656
training loss:  1.2982742259924596
test loss:  1.875217122317036
```



test loss increased to 1.8 from 1.4 training loss as expected meaning our model is doing better on test from fitting the training set

In [30]:

```
clf_l1cv = LogisticRegressionCV(
    cv=5,
    fit_intercept=True,
    multi_class='multinomial',
    penalty='l1',
    solver='saga',
    max_iter=50,
    Cs=[0.001,1],
    verbose=1
)
clf_l1cv.fit(train_img, train_lbl)
cifar_pred = clf_l1cv.predict(test_img)
score = clf_l1cv.score(test_img, test_lbl)
print('score from log cv: {}'.format(score))
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
Epoch 1, change: 1.00000000
Epoch 2, change: 0.30247099
Epoch 3, change: 0.15612592
Epoch 4, change: 0.11623720
Epoch 5, change: 0.09545401
Epoch 6, change: 0.08576925
Epoch 7, change: 0.07352968
Epoch 8, change: 0.06703851
Epoch 9, change: 0.05999550
Epoch 10, change: 0.05621588
Epoch 11, change: 0.05235974
Epoch 12, change: 0.04878032
```

```
Epoch 13, change: 0.04350183
Epoch 14, change: 0.04147405
Epoch 15, change: 0.03894039
Epoch 16, change: 0.03639959
Epoch 17, change: 0.03453178
Epoch 18, change: 0.03289401
Epoch 19, change: 0.03112194
Epoch 20, change: 0.02948586
Epoch 21, change: 0.02792373
Epoch 22, change: 0.02659860
Epoch 23, change: 0.02537358
Epoch 24, change: 0.02424722
Epoch 25, change: 0.02324387
Epoch 26, change: 0.02231609
Epoch 27, change: 0.02159214
Epoch 28, change: 0.02082675
Epoch 29, change: 0.02018458
Epoch 30, change: 0.01949564
Epoch 31, change: 0.01884952
Epoch 32, change: 0.01826136
Epoch 33, change: 0.01773619
Epoch 34, change: 0.01699985
Epoch 35, change: 0.01648719
Epoch 36, change: 0.01597205
Epoch 37, change: 0.01524046
Epoch 38, change: 0.01483776
Epoch 39, change: 0.01434918
Epoch 40, change: 0.01386519
Epoch 41, change: 0.01354282
Epoch 42, change: 0.01311676
Epoch 43, change: 0.01280477
Epoch 44, change: 0.01240429
Epoch 45, change: 0.01217610
Epoch 46, change: 0.01178393
Epoch 47, change: 0.01151838
Epoch 48, change: 0.01121196
Epoch 49, change: 0.01094951
Epoch 50, change: 0.01068041
max_iter reached after 83 seconds
/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.02202578
Epoch 3, change: 0.02053226
Epoch 4, change: 0.01939188
Epoch 5, change: 0.01791318
Epoch 6, change: 0.01735578
Epoch 7, change: 0.01594521
Epoch 8, change: 0.01544002
Epoch 9, change: 0.01462148
Epoch 10, change: 0.01409662
Epoch 11, change: 0.01350787
Epoch 12, change: 0.01289895
Epoch 13, change: 0.01243285
Epoch 14, change: 0.01193808
Epoch 15, change: 0.01166240
Epoch 16, change: 0.01130522
Epoch 17, change: 0.01081802
```

```
Epoch 18, change: 0.01062389
Epoch 19, change: 0.01034196
Epoch 20, change: 0.01004384
Epoch 21, change: 0.00971330
Epoch 22, change: 0.00951257
Epoch 23, change: 0.00928030
Epoch 24, change: 0.00909485
Epoch 25, change: 0.00887709
Epoch 26, change: 0.00870258
Epoch 27, change: 0.00848132
Epoch 28, change: 0.00830370
Epoch 29, change: 0.00815766
Epoch 30, change: 0.00812200
Epoch 31, change: 0.00793574
Epoch 32, change: 0.00782241
Epoch 33, change: 0.00762882
Epoch 34, change: 0.00759899
Epoch 35, change: 0.00738436
Epoch 36, change: 0.00737084
Epoch 37, change: 0.00724392
Epoch 38, change: 0.00713780
Epoch 39, change: 0.00705594
Epoch 40, change: 0.00691282
Epoch 41, change: 0.00691022
Epoch 42, change: 0.00675110
Epoch 43, change: 0.00670388
Epoch 44, change: 0.00654333
Epoch 45, change: 0.00654223
Epoch 46, change: 0.00644836
Epoch 47, change: 0.00637535
Epoch 48, change: 0.00623290
Epoch 49, change: 0.00618116
max_iter reached after 101 secondsEpoch 50, change: 0.00615323

/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.25071132
Epoch 3, change: 0.14780518
Epoch 4, change: 0.12042094
Epoch 5, change: 0.10153628
Epoch 6, change: 0.08667278
Epoch 7, change: 0.07639031
Epoch 8, change: 0.06819379
Epoch 9, change: 0.06252214
Epoch 10, change: 0.05651635
Epoch 11, change: 0.05216272
Epoch 12, change: 0.04744105
Epoch 13, change: 0.04458606
Epoch 14, change: 0.04172965
Epoch 15, change: 0.03940092
Epoch 16, change: 0.03715007
Epoch 17, change: 0.03487713
Epoch 18, change: 0.03325111
Epoch 19, change: 0.03190538
Epoch 20, change: 0.03030579
Epoch 21, change: 0.02863405
Epoch 22, change: 0.02797592
```

```
Epoch 23, change: 0.02666162
Epoch 24, change: 0.02568527
Epoch 25, change: 0.02485864
Epoch 26, change: 0.02393258
Epoch 27, change: 0.02330391
Epoch 28, change: 0.02250252
Epoch 29, change: 0.02167023
Epoch 30, change: 0.02092967
Epoch 31, change: 0.02044864
Epoch 32, change: 0.01974456
Epoch 33, change: 0.01921819
Epoch 34, change: 0.01854864
Epoch 35, change: 0.01808652
Epoch 36, change: 0.01743524
Epoch 37, change: 0.01691496
Epoch 38, change: 0.01642826
Epoch 39, change: 0.01584311
Epoch 40, change: 0.01533999
Epoch 41, change: 0.01486777
Epoch 42, change: 0.01445512
Epoch 43, change: 0.01402602
Epoch 44, change: 0.01360898
Epoch 45, change: 0.01326557
Epoch 46, change: 0.01285126
Epoch 47, change: 0.01262804
Epoch 48, change: 0.01231509
Epoch 49, change: 0.01204373
max_iter reached after 83 secondsEpoch 50, change: 0.01173567
```

```
/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.02405745
Epoch 3, change: 0.02333125
Epoch 4, change: 0.02111274
Epoch 5, change: 0.02031244
Epoch 6, change: 0.01908170
Epoch 7, change: 0.01789531
Epoch 8, change: 0.01730724
Epoch 9, change: 0.01653323
Epoch 10, change: 0.01590862
Epoch 11, change: 0.01538887
Epoch 12, change: 0.01463017
Epoch 13, change: 0.01435223
Epoch 14, change: 0.01396896
Epoch 15, change: 0.01325622
Epoch 16, change: 0.01300548
Epoch 17, change: 0.01272704
Epoch 18, change: 0.01229558
Epoch 19, change: 0.01187479
Epoch 20, change: 0.01160638
Epoch 21, change: 0.01131322
Epoch 22, change: 0.01111372
Epoch 23, change: 0.01081386
Epoch 24, change: 0.01053618
Epoch 25, change: 0.01031443
Epoch 26, change: 0.01008689
Epoch 27, change: 0.00982038
```

```
Epoch 28, change: 0.00958739
Epoch 29, change: 0.00937253
Epoch 30, change: 0.00919343
Epoch 31, change: 0.00898315
Epoch 32, change: 0.00879202
Epoch 33, change: 0.00865791
Epoch 34, change: 0.00847976
Epoch 35, change: 0.00831139
Epoch 36, change: 0.00816583
Epoch 37, change: 0.00799142
Epoch 38, change: 0.00783859
Epoch 39, change: 0.00765963
Epoch 40, change: 0.00752172
Epoch 41, change: 0.00736210
Epoch 42, change: 0.00723343
Epoch 43, change: 0.00713417
Epoch 44, change: 0.00698819
Epoch 45, change: 0.00689300
Epoch 46, change: 0.00681321
Epoch 47, change: 0.00661790
Epoch 48, change: 0.00654319
Epoch 49, change: 0.00641723
max_iter reached after 100 seconds
Epoch 50, change: 0.00629010
/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.25075919
Epoch 3, change: 0.16067090
Epoch 4, change: 0.12606463
Epoch 5, change: 0.09666869
Epoch 6, change: 0.08575741
Epoch 7, change: 0.07451744
Epoch 8, change: 0.06582365
Epoch 9, change: 0.05879058
Epoch 10, change: 0.05207591
Epoch 11, change: 0.04889797
Epoch 12, change: 0.04527587
Epoch 13, change: 0.04218316
Epoch 14, change: 0.03963653
Epoch 15, change: 0.03732133
Epoch 16, change: 0.03559326
Epoch 17, change: 0.03338541
Epoch 18, change: 0.03174640
Epoch 19, change: 0.03002503
Epoch 20, change: 0.02860156
Epoch 21, change: 0.02754572
Epoch 22, change: 0.02656375
Epoch 23, change: 0.02554274
Epoch 24, change: 0.02425650
Epoch 25, change: 0.02309525
Epoch 26, change: 0.02224215
Epoch 27, change: 0.02144109
Epoch 28, change: 0.02058553
Epoch 29, change: 0.01969336
Epoch 30, change: 0.01906281
Epoch 31, change: 0.01849721
Epoch 32, change: 0.01777302
```

```
Epoch 33, change: 0.01720138
Epoch 34, change: 0.01675962
Epoch 35, change: 0.01612642
Epoch 36, change: 0.01559498
Epoch 37, change: 0.01508002
Epoch 38, change: 0.01467191
Epoch 39, change: 0.01435382
Epoch 40, change: 0.01401250
Epoch 41, change: 0.01368053
Epoch 42, change: 0.01332789
Epoch 43, change: 0.01298945
Epoch 44, change: 0.01267059
Epoch 45, change: 0.01236437
Epoch 46, change: 0.01206037
Epoch 47, change: 0.01182831
Epoch 48, change: 0.01158316
Epoch 49, change: 0.01135176
max_iter reached after 82 secondsEpoch 50, change: 0.01116634
```

```
/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.02357867
Epoch 3, change: 0.02174784
Epoch 4, change: 0.02056447
Epoch 5, change: 0.01940243
Epoch 6, change: 0.01851822
Epoch 7, change: 0.01760180
Epoch 8, change: 0.01669123
Epoch 9, change: 0.01649406
Epoch 10, change: 0.01562893
Epoch 11, change: 0.01514386
Epoch 12, change: 0.01457787
Epoch 13, change: 0.01407414
Epoch 14, change: 0.01380394
Epoch 15, change: 0.01366119
Epoch 16, change: 0.01310232
Epoch 17, change: 0.01276804
Epoch 18, change: 0.01244736
Epoch 19, change: 0.01208403
Epoch 20, change: 0.01183619
Epoch 21, change: 0.01173740
Epoch 22, change: 0.01142924
Epoch 23, change: 0.01120849
Epoch 24, change: 0.01085630
Epoch 25, change: 0.01068896
Epoch 26, change: 0.01054381
Epoch 27, change: 0.01027458
Epoch 28, change: 0.01017079
Epoch 29, change: 0.00995500
Epoch 30, change: 0.00980007
Epoch 31, change: 0.00961456
Epoch 32, change: 0.00952613
Epoch 33, change: 0.00934454
Epoch 34, change: 0.00918648
Epoch 35, change: 0.00907163
Epoch 36, change: 0.00892763
Epoch 37, change: 0.00875962
```

```
Epoch 38, change: 0.00869069
Epoch 39, change: 0.00854389
Epoch 40, change: 0.00847226
Epoch 41, change: 0.00831963
Epoch 42, change: 0.00819762
Epoch 43, change: 0.00804496
Epoch 44, change: 0.00804652
Epoch 45, change: 0.00789398
Epoch 46, change: 0.00780047
Epoch 47, change: 0.00774271
Epoch 48, change: 0.00763723
Epoch 49, change: 0.00754698
max_iter reached after 100 seconds
Epoch 50, change: 0.00747941
/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.30212224
Epoch 3, change: 0.18466454
Epoch 4, change: 0.12924627
Epoch 5, change: 0.10184177
Epoch 6, change: 0.08528019
Epoch 7, change: 0.07433532
Epoch 8, change: 0.06779535
Epoch 9, change: 0.06141045
Epoch 10, change: 0.05552186
Epoch 11, change: 0.05072224
Epoch 12, change: 0.04721327
Epoch 13, change: 0.04361926
Epoch 14, change: 0.04117762
Epoch 15, change: 0.03840543
Epoch 16, change: 0.03612555
Epoch 17, change: 0.03412251
Epoch 18, change: 0.03224516
Epoch 19, change: 0.03065106
Epoch 20, change: 0.02907460
Epoch 21, change: 0.02770368
Epoch 22, change: 0.02633139
Epoch 23, change: 0.02535425
Epoch 24, change: 0.02427929
Epoch 25, change: 0.02311545
Epoch 26, change: 0.02206548
Epoch 27, change: 0.02142778
Epoch 28, change: 0.02048372
Epoch 29, change: 0.01982580
Epoch 30, change: 0.01901393
Epoch 31, change: 0.01827280
Epoch 32, change: 0.01772237
Epoch 33, change: 0.01698984
Epoch 34, change: 0.01643774
Epoch 35, change: 0.01594801
Epoch 36, change: 0.01544620
Epoch 37, change: 0.01493173
Epoch 38, change: 0.01450831
Epoch 39, change: 0.01412456
Epoch 40, change: 0.01373229
Epoch 41, change: 0.01327112
Epoch 42, change: 0.01291818
```

```
Epoch 43, change: 0.01251309
Epoch 44, change: 0.01221934
Epoch 45, change: 0.01192227
Epoch 46, change: 0.01161667
Epoch 47, change: 0.01128366
Epoch 48, change: 0.01101848
Epoch 49, change: 0.01072681
max_iter reached after 83 secondsEpoch 50, change: 0.01046742

/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.02182879
Epoch 3, change: 0.02075230
Epoch 4, change: 0.01877910
Epoch 5, change: 0.01791515
Epoch 6, change: 0.01662826
Epoch 7, change: 0.01588131
Epoch 8, change: 0.01524642
Epoch 9, change: 0.01484876
Epoch 10, change: 0.01438297
Epoch 11, change: 0.01377647
Epoch 12, change: 0.01312795
Epoch 13, change: 0.01287136
Epoch 14, change: 0.01239956
Epoch 15, change: 0.01204407
Epoch 16, change: 0.01187657
Epoch 17, change: 0.01146134
Epoch 18, change: 0.01111825
Epoch 19, change: 0.01082531
Epoch 20, change: 0.01061326
Epoch 21, change: 0.01035937
Epoch 22, change: 0.01002451
Epoch 23, change: 0.00983666
Epoch 24, change: 0.00959424
Epoch 25, change: 0.00944280
Epoch 26, change: 0.00918659
Epoch 27, change: 0.00901040
Epoch 28, change: 0.00886178
Epoch 29, change: 0.00870168
Epoch 30, change: 0.00845704
Epoch 31, change: 0.00825292
Epoch 32, change: 0.00822410
Epoch 33, change: 0.00805372
Epoch 34, change: 0.00790969
Epoch 35, change: 0.00779612
Epoch 36, change: 0.00763224
Epoch 37, change: 0.00746456
Epoch 38, change: 0.00734121
Epoch 39, change: 0.00732288
Epoch 40, change: 0.00718628
Epoch 41, change: 0.00707806
Epoch 42, change: 0.00697131
Epoch 43, change: 0.00683856
Epoch 44, change: 0.00676362
Epoch 45, change: 0.00659249
Epoch 46, change: 0.00659923
Epoch 47, change: 0.00647308
```

```
Epoch 48, change: 0.00636260
Epoch 49, change: 0.00627892
max_iter reached after 101 seconds
Epoch 50, change: 0.00623972
/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.27325539
Epoch 3, change: 0.16382063
Epoch 4, change: 0.11020753
Epoch 5, change: 0.10514343
Epoch 6, change: 0.08994700
Epoch 7, change: 0.07924341
Epoch 8, change: 0.07141656
Epoch 9, change: 0.06686981
Epoch 10, change: 0.06170163
Epoch 11, change: 0.05743224
Epoch 12, change: 0.05312570
Epoch 13, change: 0.04862381
Epoch 14, change: 0.04552447
Epoch 15, change: 0.04260601
Epoch 16, change: 0.03981114
Epoch 17, change: 0.03773883
Epoch 18, change: 0.03601146
Epoch 19, change: 0.03392721
Epoch 20, change: 0.03204628
Epoch 21, change: 0.03029247
Epoch 22, change: 0.02904952
Epoch 23, change: 0.02767225
Epoch 24, change: 0.02641756
Epoch 25, change: 0.02534062
Epoch 26, change: 0.02437481
Epoch 27, change: 0.02331912
Epoch 28, change: 0.02242921
Epoch 29, change: 0.02158799
Epoch 30, change: 0.02084376
Epoch 31, change: 0.02000973
Epoch 32, change: 0.01943051
Epoch 33, change: 0.01864216
Epoch 34, change: 0.01804888
Epoch 35, change: 0.01751602
Epoch 36, change: 0.01693876
Epoch 37, change: 0.01647766
Epoch 38, change: 0.01595294
Epoch 39, change: 0.01544272
Epoch 40, change: 0.01514569
Epoch 41, change: 0.01466480
Epoch 42, change: 0.01440451
Epoch 43, change: 0.01397738
Epoch 44, change: 0.01371017
Epoch 45, change: 0.01340087
Epoch 46, change: 0.01312382
Epoch 47, change: 0.01279800
Epoch 48, change: 0.01252541
Epoch 49, change: 0.01223925
max_iter reached after 84 secondsEpoch 50, change: 0.01199055

/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
```

```
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
Epoch 1, change: 1.00000000
Epoch 2, change: 0.02299916
Epoch 3, change: 0.02212013
Epoch 4, change: 0.02077896
Epoch 5, change: 0.01947365
Epoch 6, change: 0.01859857
Epoch 7, change: 0.01740468
Epoch 8, change: 0.01651944
Epoch 9, change: 0.01601619
Epoch 10, change: 0.01548607
Epoch 11, change: 0.01476766
Epoch 12, change: 0.01425361
Epoch 13, change: 0.01354939
Epoch 14, change: 0.01320503
Epoch 15, change: 0.01285540
Epoch 16, change: 0.01240202
Epoch 17, change: 0.01197832
Epoch 18, change: 0.01174475
Epoch 19, change: 0.01125208
Epoch 20, change: 0.01096592
Epoch 21, change: 0.01076687
Epoch 22, change: 0.01044067
Epoch 23, change: 0.01017851
Epoch 24, change: 0.00988087
Epoch 25, change: 0.00967125
Epoch 26, change: 0.00939603
Epoch 27, change: 0.00921757
Epoch 28, change: 0.00901723
Epoch 29, change: 0.00889698
Epoch 30, change: 0.00856867
Epoch 31, change: 0.00842829
Epoch 32, change: 0.00824058
Epoch 33, change: 0.00803454
Epoch 34, change: 0.00784323
Epoch 35, change: 0.00775088
Epoch 36, change: 0.00759616
Epoch 37, change: 0.00744906
Epoch 38, change: 0.00727925
Epoch 39, change: 0.00716379
Epoch 40, change: 0.00699231
Epoch 41, change: 0.00686129
Epoch 42, change: 0.00670882
Epoch 43, change: 0.00658062
Epoch 44, change: 0.00657565
Epoch 45, change: 0.00639770
Epoch 46, change: 0.00628112
Epoch 47, change: 0.00619082
Epoch 48, change: 0.00604592
Epoch 49, change: 0.00599177
max_iter reached after 101 secondsEpoch 50, change: 0.00587207

/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed: 15.3min finished
score from log cv: 0.394
```

```
/home/tom/.local/lib/python3.8/site-packages/sklearn/linear_model/_sag.py:328: C
onvergenceWarning: The max_iter was reached which means the coef_ did not conver
ge
  warnings.warn("The max_iter was reached which means "
```

In [31]:

```
from pprint import pprint
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 500, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'log2']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 50, num = 10)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
pprint(random_grid)
```

```
{'bootstrap': [True, False],
 'max_depth': [10, 14, 18, 23, 27, 32, 36, 41, 45, 50, None],
 'max_features': ['auto', 'log2'],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [100, 144, 188, 233, 277, 322, 366, 411, 455, 500]}
```

In [32]:

```
#Random Forest for CIFAR with CV
rfc = RandomForestClassifier(n_estimators=100)
rf_random = RandomizedSearchCV(estimator = rfc, param_distributions = random_gri
rf_random.fit(train_img,train_lbl)

#Random Forest CIFAR
rfc.fit(train_img, train_lbl)
```

```
Fitting 3 folds for each of 1 candidates, totalling 3 fits
[CV] END bootstrap=False, max_depth=36, max_features=log2, min_samples_leaf=1, m
in_samples_split=10, n_estimators=277; total time=  16.9s
[CV] END bootstrap=False, max_depth=36, max_features=log2, min_samples_leaf=1, m
in_samples_split=10, n_estimators=277; total time=  17.2s
[CV] END bootstrap=False, max_depth=36, max_features=log2, min_samples_leaf=1, m
in_samples_split=10, n_estimators=277; total time=  17.4s
```

Out[32]:

```
RandomForestClassifier()
```

In [33]:

```
rf_random.best_params_
```

Out[33]:

```
{'n_estimators': 277,
 'min_samples_split': 10,
 'min_samples_leaf': 1,
 'max_features': 'log2',
```

```
'max_depth': 36,
'bootstrap': False}
```

In [35]:

```
#Random Forest CV CIFAR Predictions
cv_pred = rf_random.predict(test_img)
print("CV prediction: ")
# for i in range(len(test_lbl.to_numpy())):
    # print("Prediction:", cv_pred[i], "Actual:", test_lbl.to_numpy()[i])
print("CV train accuracy: ", rf_random.score(train_img, train_lbl))
print("CV test accuracy: ", rf_random.score(test_img, test_lbl))
print("CV train loss: ", log_loss(train_lbl.to_numpy(), rf_random.predict_proba(
print("CV test loss: ", log_loss(test_lbl.to_numpy(), rf_random.predict_proba(te

#Random Forest CIFAR Predictions
print("Random Forest CIFAR Prediction:")
print("train accuracy: ", rfc.score(train_img, train_lbl))
print("test accuracy: ", rfc.score(test_img, test_lbl))
print("train loss: ", log_loss(train_lbl.to_numpy(), rfc.predict_proba(train_img
print("test loss: ", log_loss(test_lbl.to_numpy(), rfc.predict_proba(test_img),
```

```
CV prediction:
CV train accuracy:  1.0
CV test accuracy:  0.4352
CV train loss:  0.4531789746848688
CV test loss:  1.737289484521643
Random Forest CIFAR Prediction:
train accuracy:  1.0
test accuracy:  0.414
train loss:  0.3453414174584723
test loss:  1.7938040124813117
```

cv test accuracy (0.435) improved from non-cv random forest test accuracy (0.414) which is expected since hyperparameter is tuned

In [36]:

```
%%time
#Random Forest for MNIST
rfc = RandomForestClassifier(n_estimators=100)
rfc.fit(train_img_num,train_lbl_num)


#CV Random Forest MNIST (hyperparameter tuning)
rf_random = RandomizedSearchCV(estimator = rfc, param_distributions = random_gri
rf_random.fit(train_img_num,train_lbl_num)
```

```
Fitting 3 folds for each of 1 candidates, totalling 3 fits
[CV] END bootstrap=False, max_depth=36, max_features=log2, min_samples_leaf=1, m
in_samples_split=10, n_estimators=277; total time=  14.3s
[CV] END bootstrap=False, max_depth=36, max_features=log2, min_samples_leaf=1, m
in_samples_split=10, n_estimators=277; total time=  14.4s
[CV] END bootstrap=False, max_depth=36, max_features=log2, min_samples_leaf=1, m
in_samples_split=10, n_estimators=277; total time=  14.5s
CPU times: user 33.8 s, sys: 209 ms, total: 34 s
Wall time: 49.4 s
```

Out[36]:

```
RandomizedSearchCV(cv=3, estimator=RandomForestClassifier(), n_iter=1,
                   n_jobs=-1,
                   param_distributions={'bootstrap': [True, False],
                                        'max_depth': [10, 14, 18, 23, 27, 32,
                                                      36, 41, 45, 50, None],
                                        'max_features': ['auto', 'log2'],
```

```
                                                'min_samples_leaf': [1, 2, 4],
                                                'min_samples_split': [2, 5, 10],
                                                'n_estimators': [100, 144, 188, 233,
                                                                 277, 322, 366, 411,
                                                                 455, 500]},
                                   random_state=42, verbose=2)
```

In [38]:

```
#CV Random Forest MNIST Prediction
print("MNIST Predictions:")
print("CV train accuracy: ", rf_random.score(train_img_num, train_lbl_num))
print("CV test accuracy: ", rf_random.score(test_img_num, test_lbl_num))
print("CV train loss: ", log_loss(train_lbl_num.to_numpy(), rf_random.predict_pr
print("CV test loss: ", log_loss(test_lbl_num.to_numpy(), rf_random.predict_prob

#Random Forest MNIST Predictions
print()
print("train accuracy: ", rfc.score(train_img_num, train_lbl_num))
print("test accuracy: ", rfc.score(test_img_num, test_lbl_num))
print("train loss: ", log_loss(train_lbl_num.to_numpy(), rfc.predict_proba(train
print("test loss: ", log_loss(test_lbl_num.to_numpy(), rfc.predict_proba(test_im
```

```
MNIST Predictions:
CV train accuracy:  0.9999238095238095
CV test accuracy:  0.9589714285714286
CV train loss:  0.1520211715664609
CV test loss:  0.3792766237262439

train accuracy:  1.0
test accuracy:  0.9592
train loss:  0.0890278324074414
test loss:  0.31695277024898566
```

cv did not improved random forest MNIST prediction accuracy: 0.959 to 0.958

In [9]:

```
from scipy import stats
clf_xgb = xgb.XGBClassifier(verbosity=1)
param_dist = {
    'n_estimators': stats.randint(150, 1000),
    'learning_rate': stats.uniform(0.1, 0.3),
    'subsample': stats.uniform(0.3, 0.6),
    'max_depth': [6, 7, 8],
    'colsample_bytree': stats.uniform(0.5, 0.4),
    'min_child_weight': [4, 5]
}

# numFolds = 5
# kfold_5 = cross_validation.KFold(n = len(X), shuffle = True, n_folds = numFold

clf = RandomizedSearchCV(clf_xgb,
                         param_distributions = param_dist,
                         cv = 5,
                         n_iter = 1,
                         scoring = 'roc_auc',
                         error_score = 0,
                         verbose = 3,
                         n_jobs = -1)

clf.fit(train_img_num, train_lbl_num)

#Gradient Boosting on MNIST
```

```python
gbc = XGBClassifier(verbosity=1)
gbc.fit(train_img_num, train_lbl_num)
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
```

```
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
```

```
[23:04:34] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[23:04:35] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[23:04:35] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[23:04:36] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
```

```
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
```

```
[CV 1/5] END colsample_bytree=0.8729467994589013, learning_rate=0.28682642622182
747, max_depth=7, min_child_weight=5, n_estimators=253, subsample=0.552892112470
8609;, score=0.000 total time=41.0min
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
[CV 2/5] END colsample_bytree=0.8729467994589013, learning_rate=0.28682642622182
747, max_depth=7, min_child_weight=5, n_estimators=253, subsample=0.552892112470
8609;, score=0.000 total time=41.1min
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
[CV 3/5] END colsample_bytree=0.8729467994589013, learning_rate=0.28682642622182
747, max_depth=7, min_child_weight=5, n_estimators=253, subsample=0.552892112470
8609;, score=0.000 total time=41.2min
[23:45:28] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[CV 4/5] END colsample_bytree=0.8729467994589013, learning_rate=0.28682642622182
747, max_depth=7, min_child_weight=5, n_estimators=253, subsample=0.552892112470
8609;, score=0.000 total time=41.3min
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
```

```
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
[CV 5/5] END colsample_bytree=0.8729467994589013, learning_rate=0.28682642622182
747, max_depth=7, min_child_weight=5, n_estimators=253, subsample=0.552892112470
8609;, score=0.000 total time= 4.4min
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
[23:49:37] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[23:54:59] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
```

Out[9]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=4, num_parallel_tree=1,
              objective='multi:softprob', random_state=0, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=None, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=1)
```

In [39]:

```
#Gradient Boosting MNIST Predictions
cv_pred = clf.predict(test_img_num)
pred = gbc.predict(test_img_num)
print('cv test accuracy: ', accuracy_score(test_lbl_num, cv_pred))
print('non-cv test accuracy: ', accuracy_score(test_lbl_num, pred))
```

```
cv test accuracy:  0.9664
```

```
non-cv test accuracy:  0.9675428571428571
```

In [13]:
```
clf.best_params_
```

Out[13]:
```
{'colsample_bytree': 0.8729467994589013,
 'learning_rate': 0.28682642622182747,
 'max_depth': 7,
 'min_child_weight': 5,
 'n_estimators': 253,
 'subsample': 0.5528921124708609}
```

cv did not improve test accuracy compared to out-of-box XGBClassifier, but cv best params was found which is likely similar to what the model itself used.

In [16]:
```
clf_xgb_cifar = xgb.XGBClassifier(verbosity=1)
param_dist = {
    'n_estimators': stats.randint(150, 1000),
    'learning_rate': stats.uniform(0.1, 0.3),
    'subsample': stats.uniform(0.3, 0.6),
    'max_depth': [6, 7, 8],
    'colsample_bytree': stats.uniform(0.5, 0.4),
    'min_child_weight': [4, 5]
}

# numFolds = 5
# kfold_5 = cross_validation.KFold(n = len(X), shuffle = True, n_folds = numFold

clf_cifar = RandomizedSearchCV(clf_xgb_cifar,
                         param_distributions = param_dist,
                         cv = 5,
                         n_iter = 1, # you want 5 here not 25 if I understand yo
                         scoring = 'roc_auc',
                         error_score = 0,
                         verbose = 3,
                         n_jobs = -1)

clf_cifar.fit(train_img, train_lbl)

#Gradient Boosting on CIFAR
gbc_cifar = XGBClassifier(verbosity=1)
gbc_cifar.fit(train_img, train_lbl)
```

```
Fitting 5 folds for each of 1 candidates, totalling 5 fits
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
```

```
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
[10:43:58] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[10:43:59] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[10:43:59] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[10:44:00] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
[CV 4/5] END colsample_bytree=0.756731368962299, learning_rate=0.205880981432285
9, max_depth=7, min_child_weight=4, n_estimators=601, subsample=0.55253027691705
37;, score=0.000 total time=104.7min
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
```

```
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
[CV 1/5] END colsample_bytree=0.756731368962299, learning_rate=0.205880981432285
9, max_depth=7, min_child_weight=4, n_estimators=601, subsample=0.55253027691705
37;, score=0.000 total time=104.8min
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
[CV 3/5] END colsample_bytree=0.756731368962299, learning_rate=0.205880981432285
9, max_depth=7, min_child_weight=4, n_estimators=601, subsample=0.55253027691705
37;, score=0.000 total time=104.9min
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
[CV 2/5] END colsample_bytree=0.756731368962299, learning_rate=0.205880981432285
9, max_depth=7, min_child_weight=4, n_estimators=601, subsample=0.55253027691705
37;, score=0.000 total time=104.9min
[12:28:24] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[CV 5/5] END colsample_bytree=0.756731368962299, learning_rate=0.205880981432285
9, max_depth=7, min_child_weight=4, n_estimators=601, subsample=0.55253027691705
37;, score=0.000 total time=12.4min
/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_validatio
n.py:696: UserWarning: Scoring failed. The score on this train-test partition fo
r these parameters will be set to 0. Details:
Traceback (most recent call last):
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/model_selection/_va
lidation.py", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 199, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true,
  File "/home/tom/.local/lib/python3.8/site-packages/sklearn/metrics/_scorer.p
y", line 328, in _score
    raise ValueError("{0} format is not supported".format(y_type))
ValueError: multiclass format is not supported

  warnings.warn(
/home/tom/.local/lib/python3.8/site-packages/xgboost/sklearn.py:1146: UserWarnin
g: The use of label encoder in XGBClassifier is deprecated and will be removed i
n a future release. To remove this warning, do the following: 1) Pass option use
_label_encoder=False when constructing XGBClassifier object; and 2) Encode your
labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
[12:40:40] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
[12:56:01] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the defau
lt evaluation metric used with the objective 'multi:softprob' was changed from
'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the
old behavior.
```

Out[16]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.300000012, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=100, n_jobs=4, num_parallel_tree=1,
              objective='multi:softprob', random_state=0, reg_alpha=0,
              reg_lambda=1, scale_pos_weight=None, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=1)
```

In [25]:

```
from sklearn.metrics import log_loss
#Gradient Boosting CIFAR Predictions
cv_pred = clf_cifar.predict(test_img)
cv_pred_proba = clf_cifar.predict_proba(test_img)
pred = gbc_cifar.predict(test_img)
pred_proba = gbc_cifar.predict_proba(test_img)
print('cv accuracy score: ', accuracy_score(test_lbl, cv_pred))
print('accuracy score: ', accuracy_score(test_lbl, pred))

# print(pred)

# Loss Stats
print('cv log loss: ', log_loss(test_lbl, cv_pred_proba, normalize=False))
print('log loss: ', log_loss(test_lbl, pred_proba, normalize=False))
```

```
cv accuracy score:  0.4768
accuracy score:  0.4716
cv log loss:  4056.2402420630533
log loss:  4044.918773193378
```

In [40]:

```
clf_cifar.best_params_
```

Out[40]:

```
{'colsample_bytree': 0.756731368962299,
 'learning_rate': 0.2058809814322859,
 'max_depth': 7,
 'min_child_weight': 4,
```

```
 'n_estimators': 601,
 'subsample': 0.5525302769170537}
```

cv improve accuracy about 0.005 (0.5%). Best params are shown above.

```
from pathlib import Path
import requests

DATA_PATH = Path("data")
PATH = DATA_PATH / "mnist"

PATH.mkdir(parents=True, exist_ok=True)

URL = "https://github.com/pytorch/tutorials/raw/master/_static/"
FILENAME = "mnist.pkl.gz"

if not (PATH / FILENAME).exists():
        content = requests.get(URL + FILENAME).content
        (PATH / FILENAME).open("wb").write(content)


import torch
import pickle
import gzip

with gzip.open((PATH / FILENAME).as_posix(), "rb") as f:
        ((x_train, y_train), (x_valid, y_valid), _) = pickle.load(f,
encoding="latin-1")

x_train, y_train, x_valid, y_valid = map(
    torch.tensor, (x_train, y_train, x_valid, y_valid)
)

n, c = x_train.shape
print(x_train, y_train)
print(x_train.shape)
print(y_train.min(), y_train.max())

import torch.nn.functional as F
from torch import nn

from torch.utils.data import DataLoader
from torch.utils.data import TensorDataset
from torch import optim
bs = 64
epochs = 15
lr = .1
train_ds = TensorDataset(x_train, y_train)
train_dl = DataLoader(train_ds, batch_size=bs, shuffle=True)

valid_ds = TensorDataset(x_valid, y_valid)
valid_dl = DataLoader(valid_ds, batch_size=bs * 2)
loss_func = F.cross_entropy
def accuracy(out, yb):
```

```
    preds = torch.argmax(out, dim=1)
    return (preds == yb).float().mean()

class Lambda(nn.Module):
    def __init__(self, func):
        super().__init__()
        self.func = func

    def forward(self, x):
        return self.func(x)

def preprocess(x, y):
    return x.view(-1, 1, 28, 28), y


class WrappedDataLoader:
    def __init__(self, dl, func):
        self.dl = dl
        self.func = func

    def __len__(self):
        return len(self.dl)

    def __iter__(self):
        batches = iter(self.dl)
        for b in batches:
            yield (self.func(*b))

def get_data(train_ds, valid_ds, bs):
    return (
        DataLoader(train_ds, batch_size=bs, shuffle=True),
        DataLoader(valid_ds, batch_size=bs * 2),
    )
def loss_batch(model, loss_func, xb, yb, opt=None):
    loss = loss_func(model(xb), yb)

    if opt is not None:
        loss.backward()
        opt.step()
        opt.zero_grad()

    return loss.item(), len(xb)

import numpy as np

def fit(epochs, model, loss_func, opt, train_dl, valid_dl):
    for epoch in range(epochs):
        model.train()
        for xb, yb in train_dl:
```

```
            loss_batch(model, loss_func, xb, yb, opt)

        model.eval()
        with torch.no_grad():
            losses, nums = zip(
                *[loss_batch(model, loss_func, xb, yb) for xb, yb in
valid_dl]
            )
        val_loss = np.sum(np.multiply(losses, nums)) / np.sum(nums)

        print(epoch, val_loss)


train_dl, valid_dl = get_data(train_ds, valid_ds, bs)
train_dl = WrappedDataLoader(train_dl, preprocess)
valid_dl = WrappedDataLoader(valid_dl, preprocess)

model = nn.Sequential(
    nn.Conv2d(1, 16, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(16, 16, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(16, 16, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(16, 10, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.AdaptiveAvgPool2d(1),
    Lambda(lambda x: x.view(x.size(0), -1)),
)

opt = optim.SGD(model.parameters(), lr=.01, momentum=0.9)
fit(epochs, model, loss_func, opt, train_dl, valid_dl)

tensor([[0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        ...,
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.]]) tensor([5, 0, 4,  ..., 8, 4,
8])
torch.Size([50000, 784])
tensor(0) tensor(9)
0 0.4013837484836578
1 0.22418380403518676
2 0.1759802500784397
3 0.14104989574551582
4 0.119158509349823
5 0.10803587388694286
6 0.0987531075567007
```

```
7 0.09595525956153869
8 0.09110467154532671
9 0.09254756705909968
10 0.08610441810190678
11 0.0855987502887845
12 0.08809963461384177
13 0.07974340834617615
14 0.08031095509082078
```

```
pred_list = []
sum = 0
for xb, yb in train_dl:
    ypred = model(xb)
    sum = sum + accuracy(ypred,yb)
acc = sum/len(train_dl)
print("Train Accuracy:",acc)
pred_list = []
sum = 0
for xb, yb in valid_dl:
    ypred = model(xb)
    sum = sum + accuracy(ypred,yb)
acc = sum/len(valid_dl)
print("Test Accuracy:",acc)
```

```
Train Accuracy: tensor(0.9806)
Test Accuracy: tensor(0.9773)
```

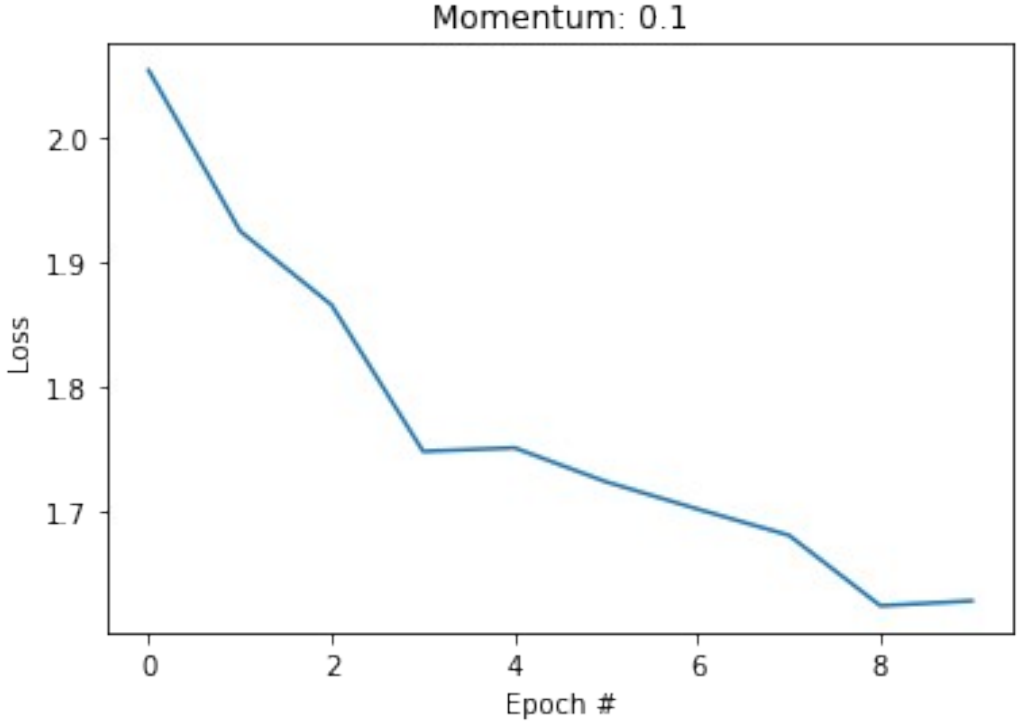
We tried many different learning rates and momentums in order to try to tune accuracy of model. First off all, for all trials we found that using momentum increased the training and testing accuracy. The model performed the worst at momentum = 0, and then slowly got better as momentum went to .9 Additonally, we tried to tune the learning rate trying values .025, .05. , .1, .15, .2. We found that the model performed the best with a learning rate of .1. Lastly, we tuned the depth of the tree and found that adding one extra layer to the NN produced the best accuracy.

```
import numpy as np
import sklearn.datasets
import torch
import torch.nn.functional as F
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from torch import nn
from torch import optim
from torch.utils.data import DataLoader
from torch.utils.data import TensorDataset

dataset = sklearn.datasets.fetch_openml("CIFAR_10_small", cache=True)
X_data = dataset.data.values
Y_data = dataset.target.values.to_numpy(dtype=np.int)
x_train, x_valid, y_train, y_valid = train_test_split(X_data, Y_data,
test_size= .25, random_state=0)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_valid = scaler.transform(x_valid)

x_train = torch.FloatTensor(x_train)
x_valid = torch.FloatTensor(x_valid)
y_train = torch.LongTensor(y_train)
y_valid = torch.LongTensor(y_valid)



n, c = x_train.shape
print(x_train, y_train)
print(x_train.shape)
print(y_train.min(), y_train.max())



bs = 64
epochs = 10
lr = .1

train_ds = TensorDataset(x_train, y_train)
train_dl = DataLoader(train_ds, batch_size=bs, shuffle=True)

valid_ds = TensorDataset(x_valid, y_valid)
valid_dl = DataLoader(valid_ds, batch_size=bs * 2)

loss_func = F.cross_entropy

def accuracy(out, yb):
    preds = torch.argmax(out, dim=1)
    return (preds == yb).float().mean()
```

```
class Lambda(nn.Module):
    def __init__(self, func):
        super().__init__()
        self.func = func

    def forward(self, x):
        return self.func(x)

def preprocess(x, y):
    return x.view(-1, 3, 32, 32), y


class WrappedDataLoader:
    def __init__(self, dl, func):
        self.dl = dl
        self.func = func

    def __len__(self):
        return len(self.dl)

    def __iter__(self):
        batches = iter(self.dl)
        for b in batches:
            yield (self.func(*b))

def get_data(train_ds, valid_ds, bs):
    return (
        DataLoader(train_ds, batch_size=bs, shuffle=True),
        DataLoader(valid_ds, batch_size=bs * 2),
    )
def loss_batch(model, loss_func, xb, yb, opt=None):
    loss = loss_func(model(xb), yb)

    if opt is not None:
        loss.backward()
        opt.step()
        opt.zero_grad()

    return loss.item(), len(xb)


def fit(epochs, model, loss_func, opt, train_dl, valid_dl):
    x_graph=[]
    y_graph=[]
    for epoch in range(epochs):
        model.train()
        for xb, yb in train_dl:
            loss_batch(model, loss_func, xb, yb, opt)
```

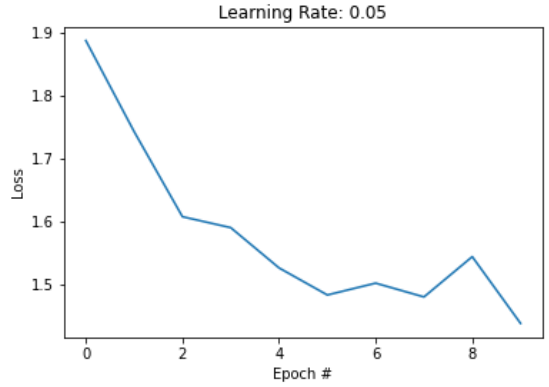
```
        model.eval()
        with torch.no_grad():
            losses, nums = zip(
                *[loss_batch(model, loss_func, xb, yb) for xb, yb in
valid_dl]
            )
        val_loss = np.sum(np.multiply(losses, nums)) / np.sum(nums)
        x_graph.append(epoch)
        y_graph.append(val_loss)
        print(epoch, val_loss)
    print(x_graph)
    print(y_graph)
    plt.plot(x_graph,y_graph)
    plt.xlabel('Epoch #')
    plt.ylabel('Loss')
    plt.title("Momentum: "+str(0.1))
    plt.savefig("m_5")
    plt.show()



train_dl, valid_dl = get_data(train_ds, valid_ds, bs)
train_dl = WrappedDataLoader(train_dl, preprocess)
valid_dl = WrappedDataLoader(valid_dl, preprocess)

model = nn.Sequential(
    nn.Conv2d(3, 16, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(16, 16, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(16, 10, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.AdaptiveAvgPool2d(1),
    Lambda(lambda x: x.view(x.size(0), -1)),
)

opt = optim.SGD(model.parameters(), lr=lr, momentum=.1)
fit(epochs, model, loss_func, opt, train_dl, valid_dl)

tensor([[-1.0693e+00, -1.1165e+00, -1.1733e+00,  ..., -7.0020e-01,
         -1.0775e+00, -4.4095e-01],
        [ 1.3308e+00,  1.3271e+00,  1.3192e+00,  ...,  1.1303e+00,
          1.1074e+00,  1.0859e+00],
        [ 1.5899e+00,  1.6447e+00,  1.4438e+00,  ...,  1.8533e+00,
          1.8408e+00,  1.8115e+00],
        ...,
        [-2.3747e-01,  1.7763e-03,  2.1141e-01,  ..., -1.1617e+00,
         -1.0164e+00, -4.1072e-01],
        [-7.3820e-02, -1.2029e-02,  5.9096e-02,  ..., -3.1563e-01,
         -2.5243e-01, -3.5025e-01],
```

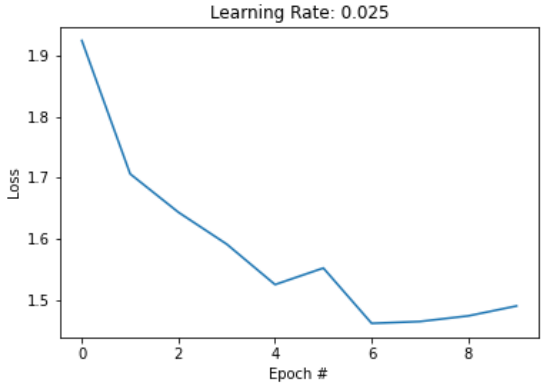
```
        [ 6.4896e-01,  3.0550e-01,  3.3603e-01,  ...,  1.7918e+00,
          1.8102e+00,  1.7662e+00]]) tensor([6, 8, 9,  ..., 4, 2, 8])
torch.Size([15000, 3072])
tensor(0) tensor(9)
0 2.052912870788574
1 1.924482466506958
2 1.8654630184173584
3 1.748558305168152
4 1.7513776527404785
5 1.7242539695739747
6 1.7024953935623168
7 1.681537672138214
8 1.625237345123291
9 1.6291268865585327
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[2.052912870788574, 1.924482466506958, 1.8654630184173584,
1.748558305168152, 1.7513776527404785, 1.7242539695739747,
1.7024953935623168, 1.681537672138214, 1.625237345123291,
1.6291268865585327]
```

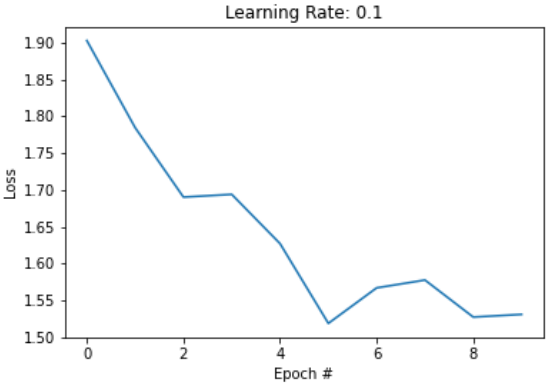


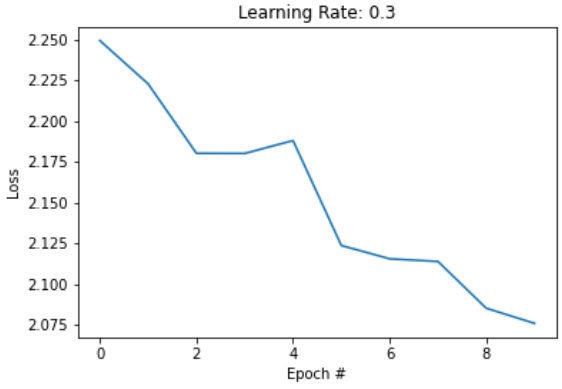

```
pred_list = []
sum = 0
for xb, yb in train_dl:
    ypred = model(xb)
    sum = sum + accuracy(ypred,yb)
acc = sum/len(train_dl)
print(acc)
```

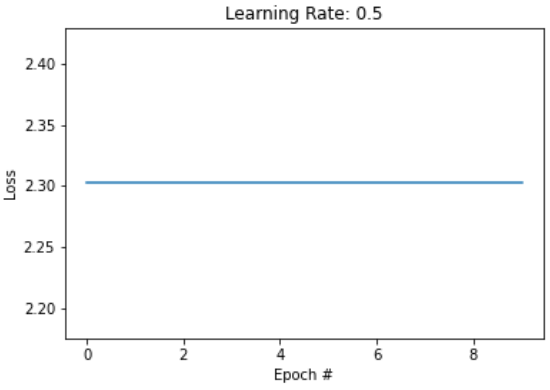
Learning Rate vs Epoch #





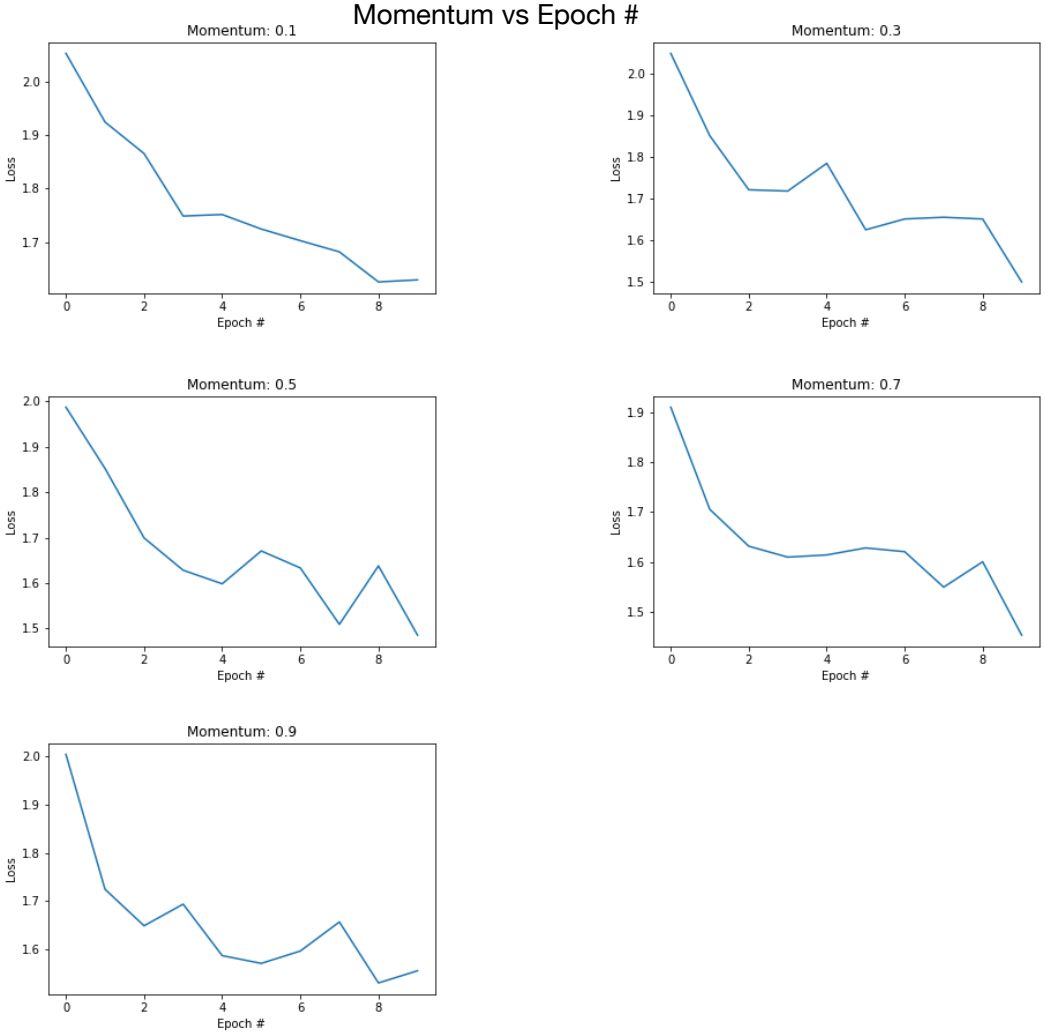






There is a strong correlation between an increase in learning rate and rate of change regarding loss. As learning rate increases, there is a smaller decrease in the loss as the number of epochs increases.

Momentum vs Epoch #



Changing momentum does not have a significant effect on the loss as the number of epochs increase, with a constant learning rate of 0.1. The rate of change regarding momentum as it changes is about the same from 0.1 to 0.9.

The original CNN had a depth of 3. After a fourth layer was added, loss didn't decrease significantly.