IT581 Adversarial Machine Learning

Lab Assignment - 02

Group 05

2021 17013: Shaikh Faizan Ahmed
2021 17014: Sonam Bharti

Question 1.

Find the "Minimum L_2 Norm" adverserial attack to a two-class linear classier by getting optimal x, solving:

$$maximize_x ||x - x_0||^2$$

subject to

$$w^T x + w_0 = 0$$

Answer

Using Lagrange multiplier of the constrained optimization is given by:

$$\mathcal{L}(x, \lambda) = \frac{1}{2}||x - x_0||^2 + \lambda(w^T x + w_0). \tag{1}$$

The solution of the optimization is the saddle point (x^*, λ^*) such that $\nabla x\mathcal{L} = 0$ and $\nabla\lambda\mathcal{L} = 0$.
Taking derivative of equation (1) with respect to x and then to λ yields:

$$\nabla x\mathcal{L} = x - x_0 + \lambda w = 0, \tag{2}$$

$$\nabla\lambda\mathcal{L} = w^T x + w_0 = 0, \tag{3}$$

Multiplying the equation (2) by $w^T yields$:

$$w^T(x - x_0) + \lambda w^T w = 0$$

$$\Rightarrow w^T x - w^T x_0 + \lambda||w||^2 = 0$$

$$\Rightarrow -w_0 - w^T x_0 + \lambda||w||^2 = 0$$

.
Thus, the optimal λ is:

$$\lambda^* = \frac{(w^T x_0 + w_0)}{||w||^2}. \tag{4}$$

Correspondingly, the optimal x is:

$$x^* = x_0 - \lambda^* w = x_0 - (\frac{w^T x_0 + w_0}{||w||})\frac{w}{||w||_2}. \tag{5}$$

Question 2.

The MNIST database of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on pre-processing and formatting.
A. Perform Linear Discriminant Analysis (LDA) on the MNIST dataset* for binary classification and find weights and bias.
a. Plot confusion matrix and accuracy.
B. Perform Minimum l2 Norm adversarial attack to set of two digits (eg. 1 and 7, 3 and 8) and get updated test dataset and try to predict with the model used in A.
a. Plot confusion matrix and accuracy.
b. Plot 10 misclassified digits as image.
Comment your observation.

Answer

A.

code:

```python
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
from pylab import *

import random

from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

mnist_train = pd.read_csv("mnist_train.csv")

np.shape(mnist_train)

mnist_train.head()

mnist_train = mnist_train.loc[mnist_train['label'].isin([3, 8])]
X = mnist_train.iloc[:,1:785]
y = mnist_train.iloc[:,0]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.40, random_state=0)

lda = LDA(n_components=1)
X_train_r2 = lda.fit(X_train, y_train)
y_pred = lda.predict(X_test)
print("Accuracy before attack: ",accuracy_score(y_test, y_pred))
print(y_pred.shape)

print(
    f"Classification report for classifier {lda}:\n"
    f"{metrics.classification_report(y_test, y_pred)}\n"
)

w0 = lda.intercept_
w = lda.coef_
print(f"Bias: \n{w0}")
print(f"\n\nWeights: \n{w}")

disp = metrics.ConfusionMatrixDisplay.from_predictions(y_test, y_pred)
disp.figure_.suptitle("Confusion Matrix")
print(f"Confusion matrix:\n{disp.confusion_matrix}")

plt.show()
```

Listing 1: Question 2(A)

Output:

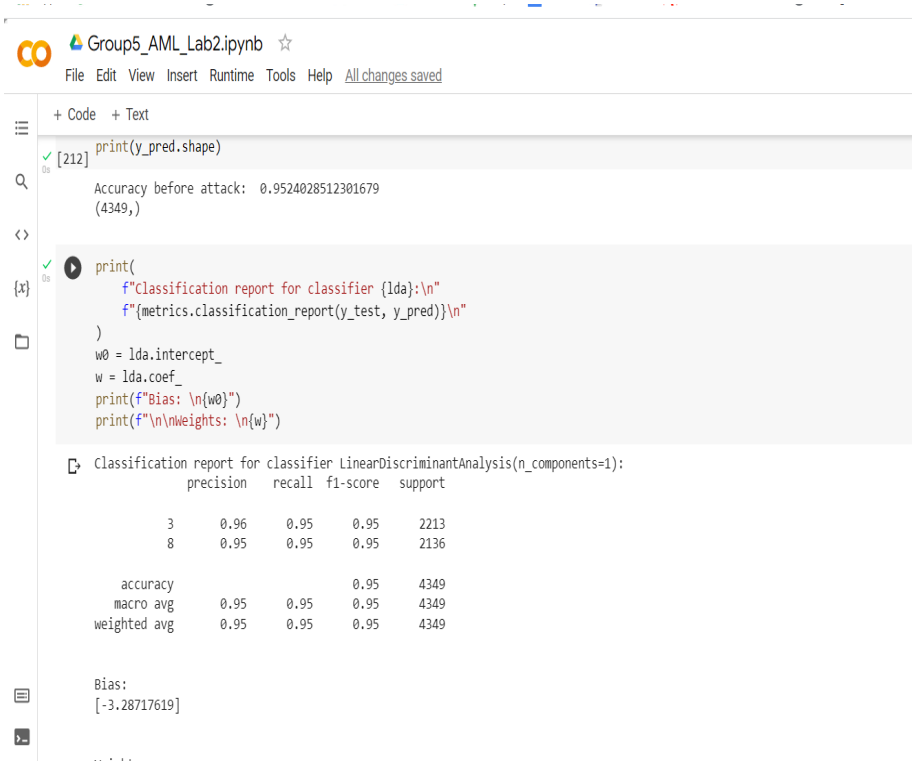


Figure 1: 2(A). Accuray

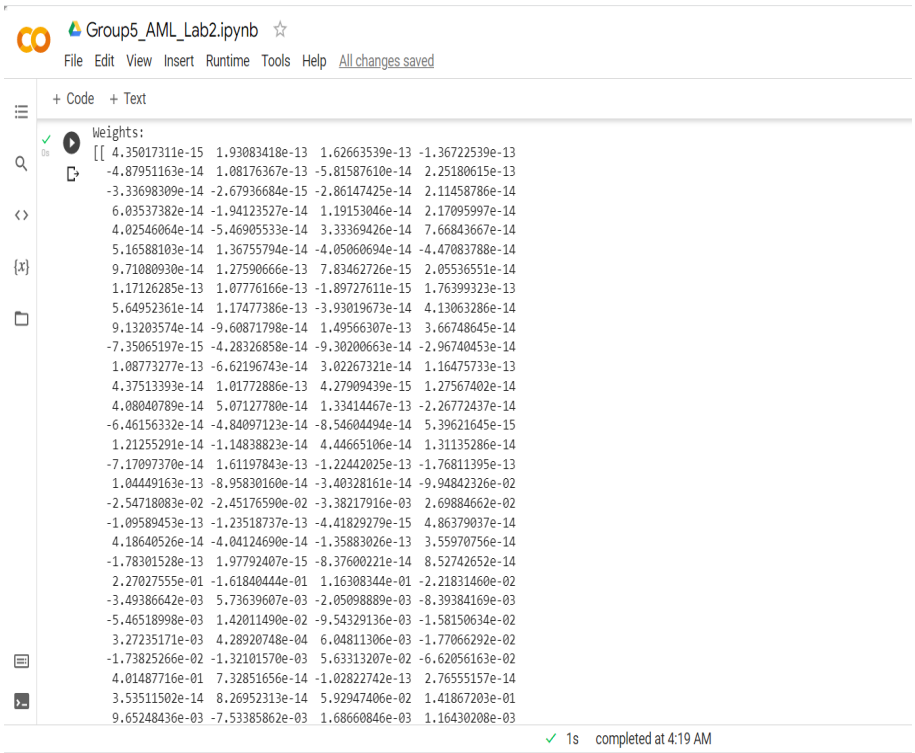


Figure 2: 2(A). Weights

Definition:

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. It is used to visualize important predictive analytics like recall, specificity, accuracy, and precision.

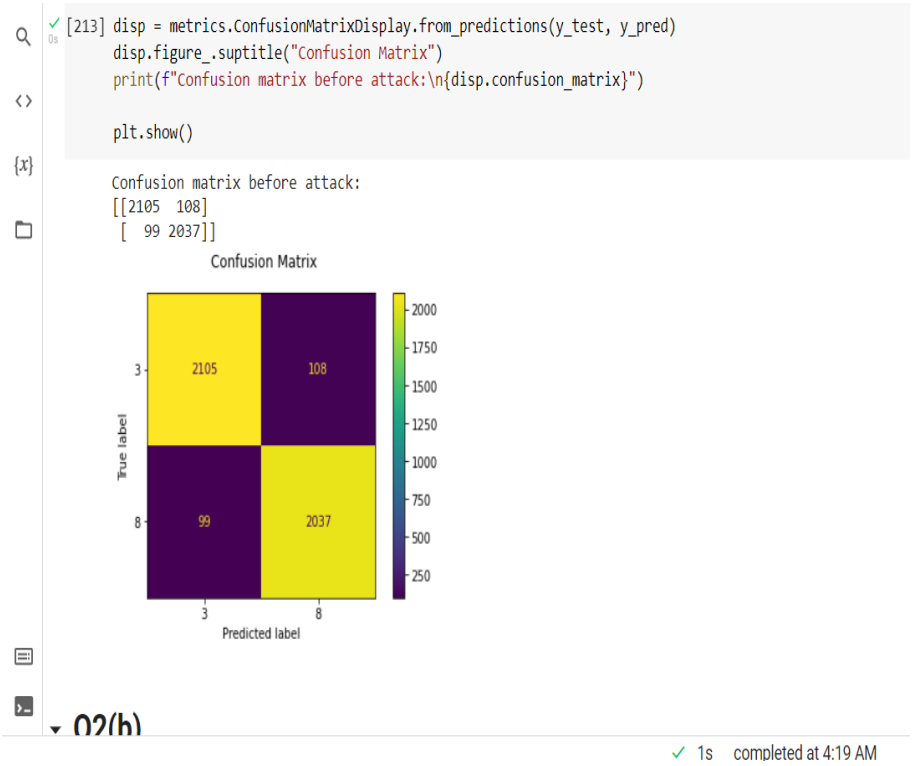


Figure 3: 2(A). Confusion Matrix before attack

Bias is a phenomenon that skews the result of an algorithm in favor or against an idea. Bias is considered a systematic error that occurs in the machine learning model itself due to incorrect assumptions in the ML process. It is defined as the intercept of the function.
Whereas, Weights control the signal (or the strength of the connection) between two neurons. In other words, a weight decides how much influence the input will have on the output. It is defined as the coefficient.

Observation / Justification:

We have imported the **MNIST** dataset from "sklearn library". Then we split our train data and test data into 6:4 ratio as per asked. Then we applied LDA algorithm to train and testing the MNIST dataset. Then we calculated the accuracy of this algorithm by using accuracy_score() function and the accuracy rate over this dataset is approximately 95%.

B.a

code:

```
X_test_final = X_test.to_numpy(dtype = 'uint8')

X_attack=X_test_final-((X_test_final@np.transpose(w)+w0)@w)/np.linalg.norm(w)
print(X_attack.shape)

Y_attack=lda.predict(X_attack)
print(Y_attack.shape)

print("Accuracy after attack: ",accuracy_score(y_test, Y_attack))

print(f"Classification report for classifier {lda}:\n"
      f"{metrics.classification_report(y_test, Y_attack)}\n")

disp = metrics.ConfusionMatrixDisplay.from_predictions(y_test, Y_attack)
disp.figure_.suptitle("Confusion Matrix")
print(f"Confusion matrix:\n{disp.confusion_matrix}")

plt.show()
```

Listing 2: Question 2(B).a

In this part we have to implement adverserial minimum norm attack over binary classifier. We tried to change 3 to 8 using minimum norm attack formula. And calculated accuracy score and Confusion Matrix after the attack.

Figure 4: 2(B).(a). Accuracy after attack

Observation / Justification:

After the minimum L_2 norm attack on the model, we observe the accuracy decreases to 4.7% for binary classifier. Also we can observe the changes in the output of confusion matrix.

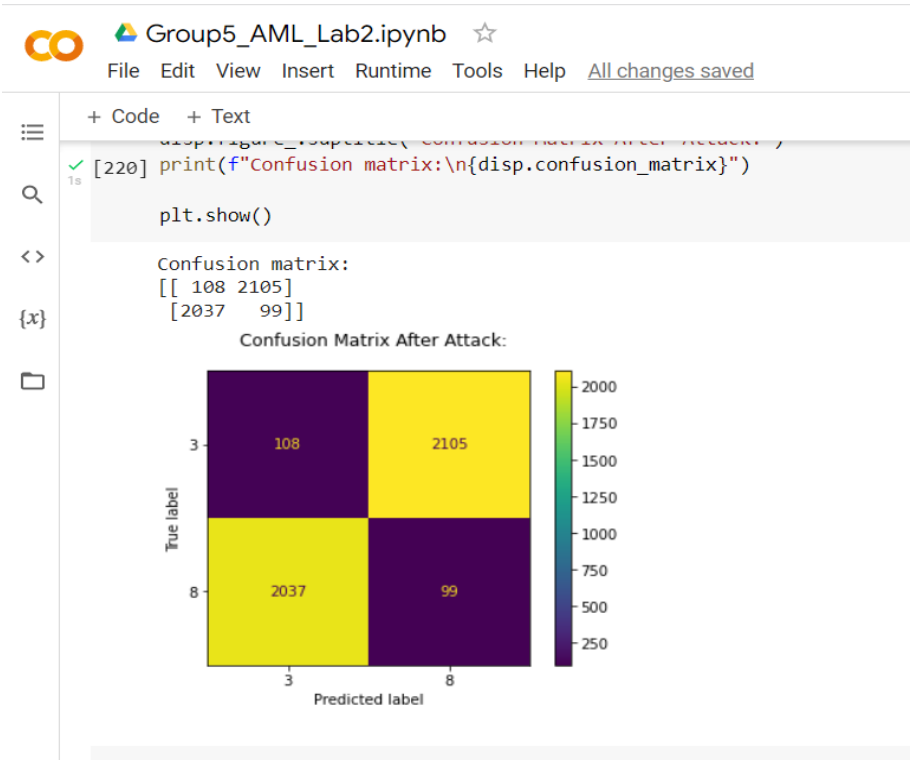


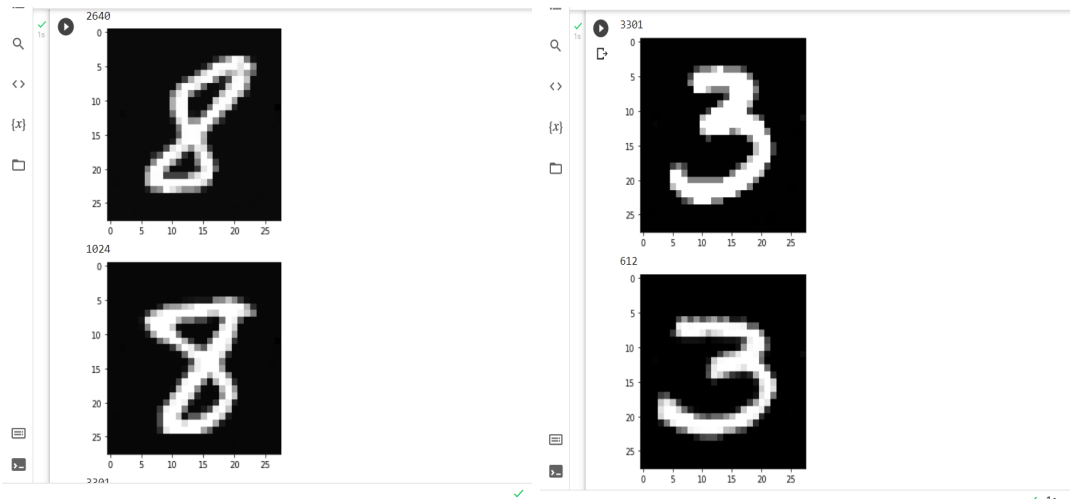
Figure 5: 2(B).(a). Confusion Matrix After attack

B.b

code:

```
X_attack_res=X_attack.reshape(X_attack.shape[0],28,28)
for i in range(0,10):
    s=random.randint(0,X_attack.shape[0])
    print(s)
    plt.imshow(X_attack_res[s])
    plt.show()
```

Listing 3: Question 2(B).b



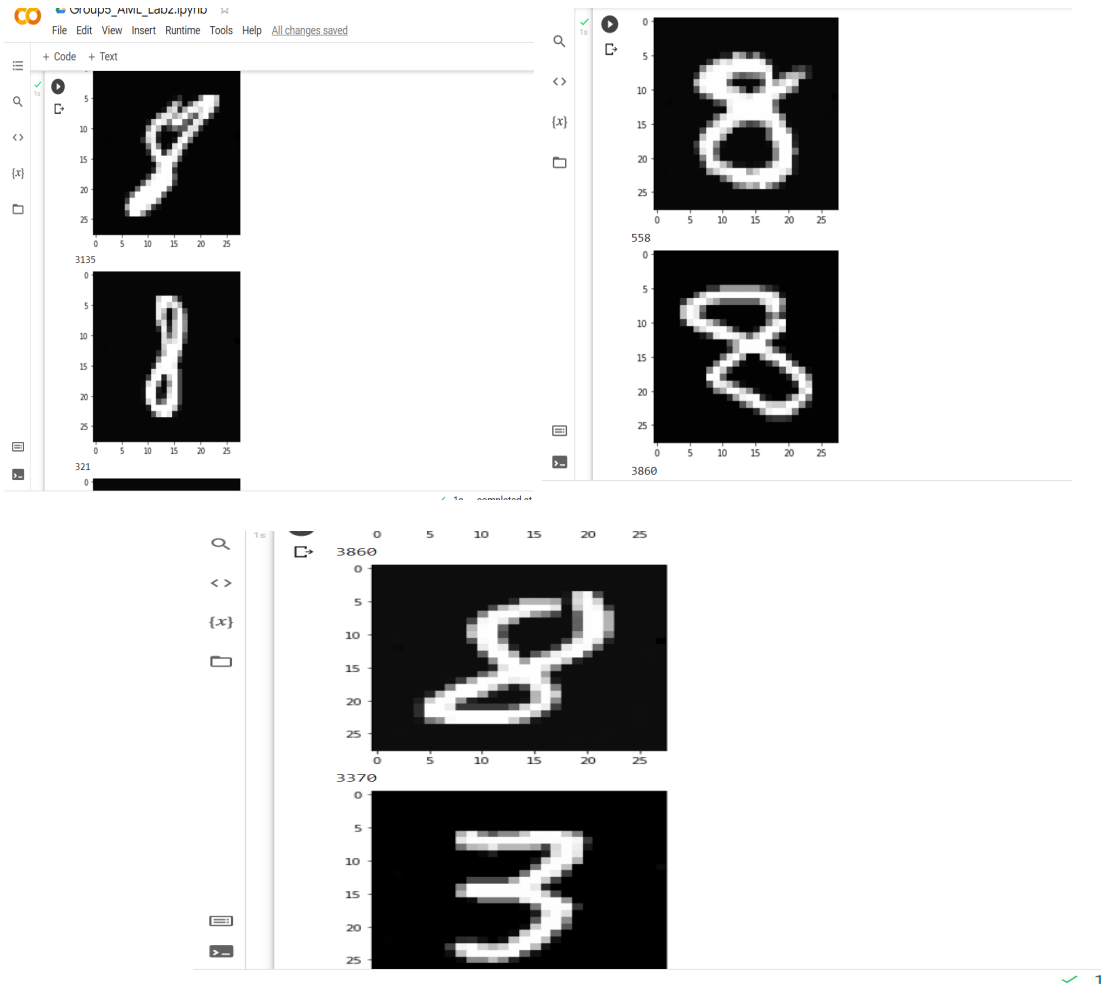


Figure 6: 2(B).(b). Image of misclassified digits

Conclusion:

We can see that after minimum L_2 norm attack most of the digit "3" is classified as "8" in 10 printed values.