DL Assignment 3 Q3 Report

Tejas Oberoi 2017367 Group 18

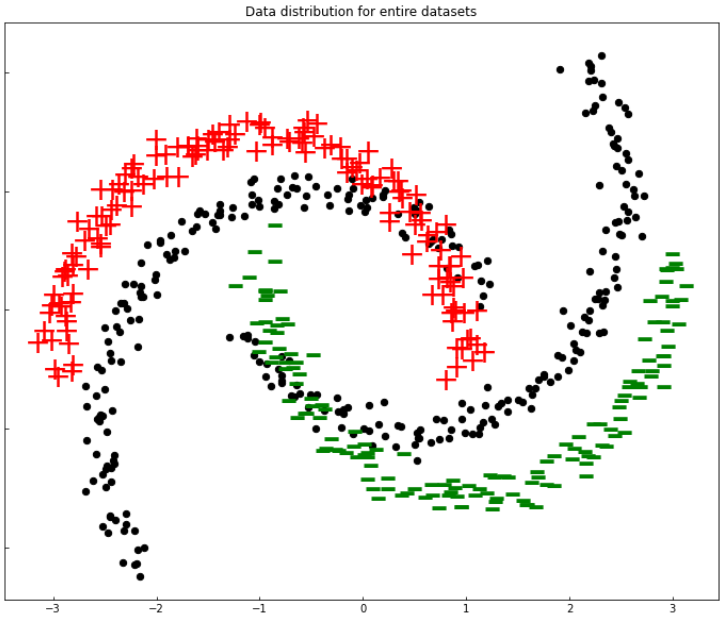
Part 1)

1. Data

Generated Distribution for complete source and target domain dataset. Each domain has 300 points wherein each label has 150 points. The unlabelled points belong to the target domain while the labelled points(+,-) belong to the source domain.

Steps mentioned in the paper were
As the source sample S, we generate a lower moon and an upper moon labeled 0 and 1 respectively, each of which containing 150 examples. The target sample T is obtained by the following procedure: (1) we generate a sample S0 the same way S has been generated; (2) we rotate each example by 35◦; and (3) we remove all the labels. Thus, T contains 300 unlabeled examples.

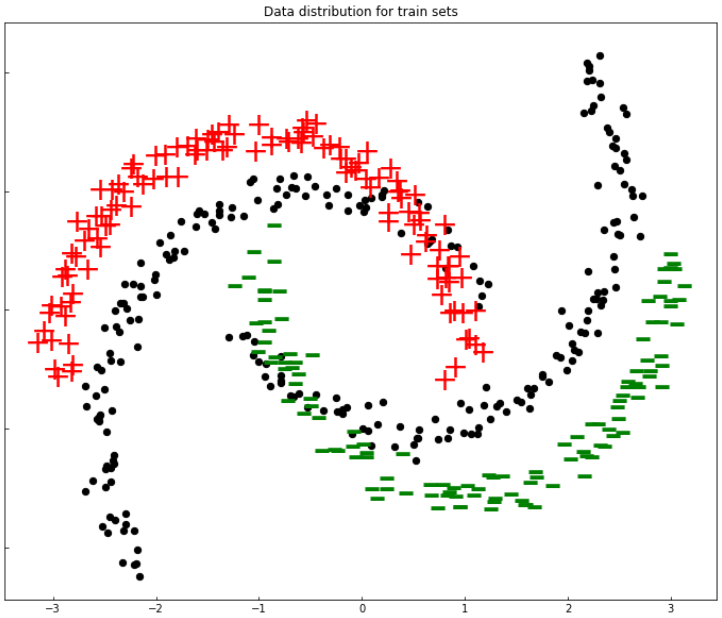
Distribution of entire dataset:



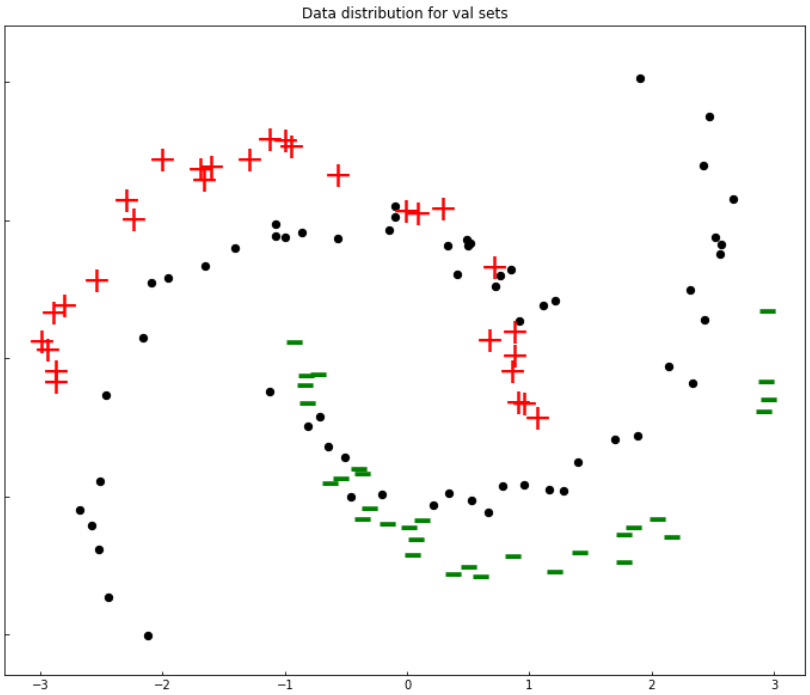
Dataset of each domain was split using an 80:20 train test split with shuffle activated.
Source Train set : 240 samples from source domain
Target Train set : 240 unlabelled samples from target domain

Source Test set: 60 samples from source domain,
Target Test set: 60 samples from target domain

Distribution of Train set:



Distribution of test set:



2. Training Details

“Shallow_Dann” architecture was replicated in pytorch as mentioned in the paper.
Feature generator: a hidden layer of 15 neurons, a sigmoid activation function
Domain Classifier: Softmax
Target Classifier: Softmax

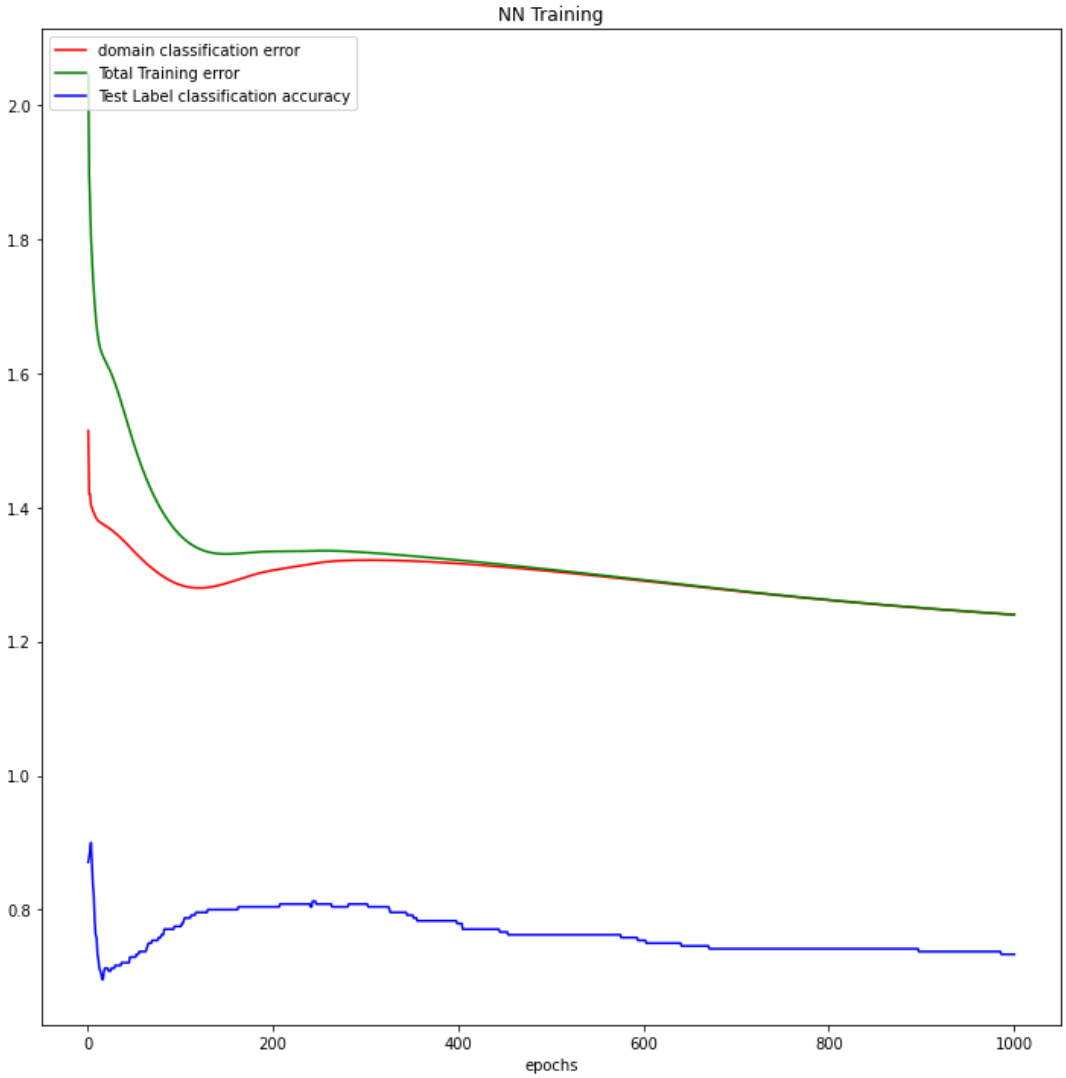
Batch size : 30
Epochs: 1000
Optimizer : Adam
Learning Rate : 0.0075
Loss Function: Negative Loss Likelihood

Pytorch Architecture used for both DANN and NN:

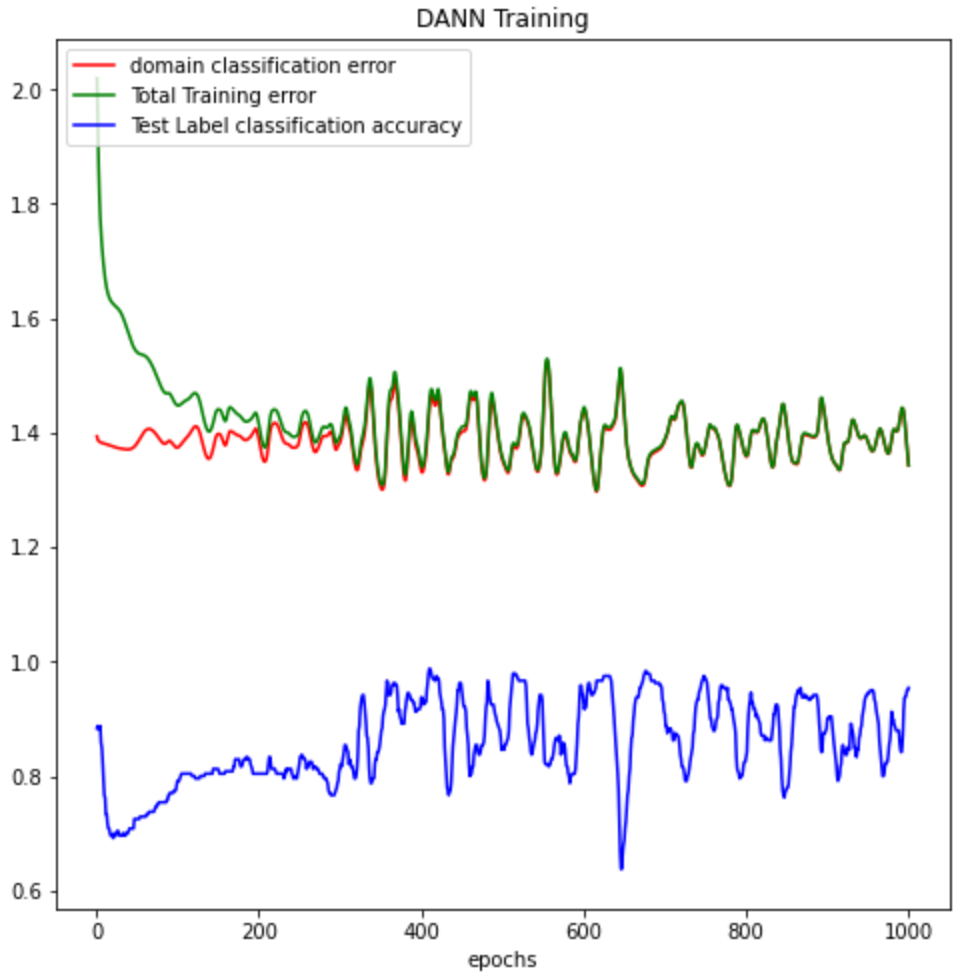
```
DANN(
  (feature): Sequential(
    (0): Linear(in_features=2, out_features=15, bias=True)
```

```
    (1): Sigmoid()
  )
  (lc): Sequential(
    (0): Linear(in_features=15, out_features=2, bias=True)
    (1): LogSoftmax()
  )
  (dc): Sequential(
    (0): Linear(in_features=15, out_features=2, bias=True)
    (1): LogSoftmax()
  )
)
```

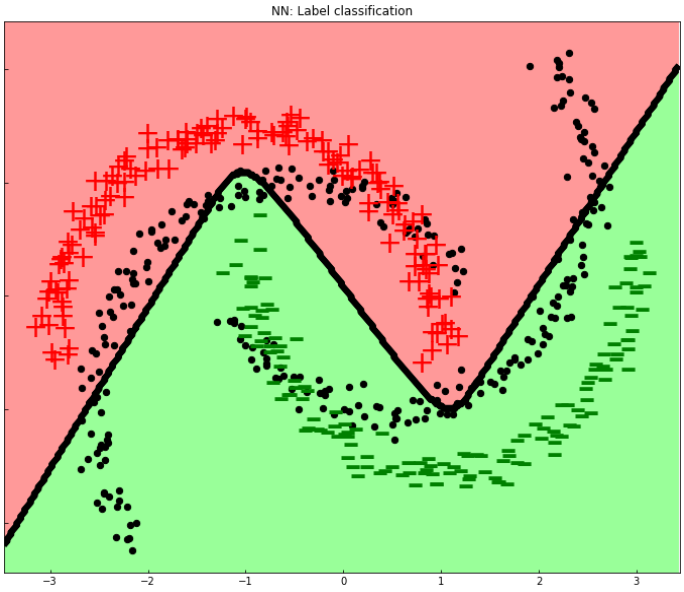
Note: Domain Classification error is for the samples in the training set.
NN Training Plot:



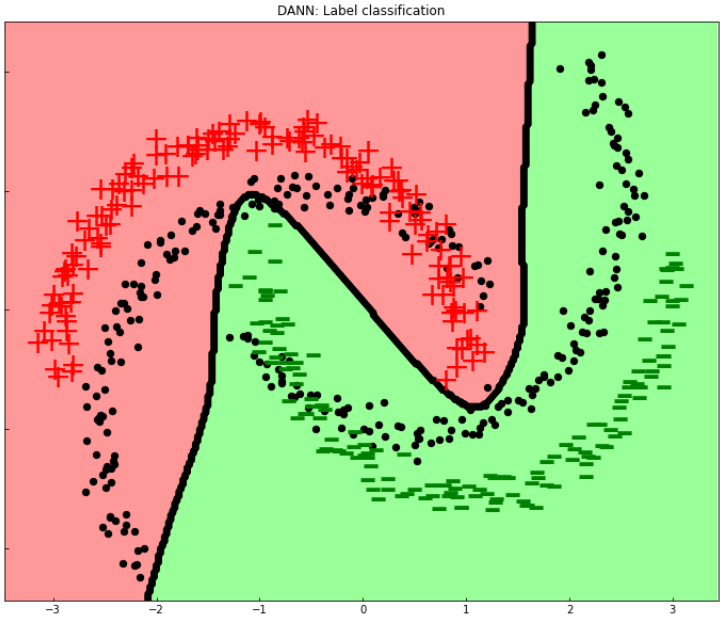
DANN Training Plot :



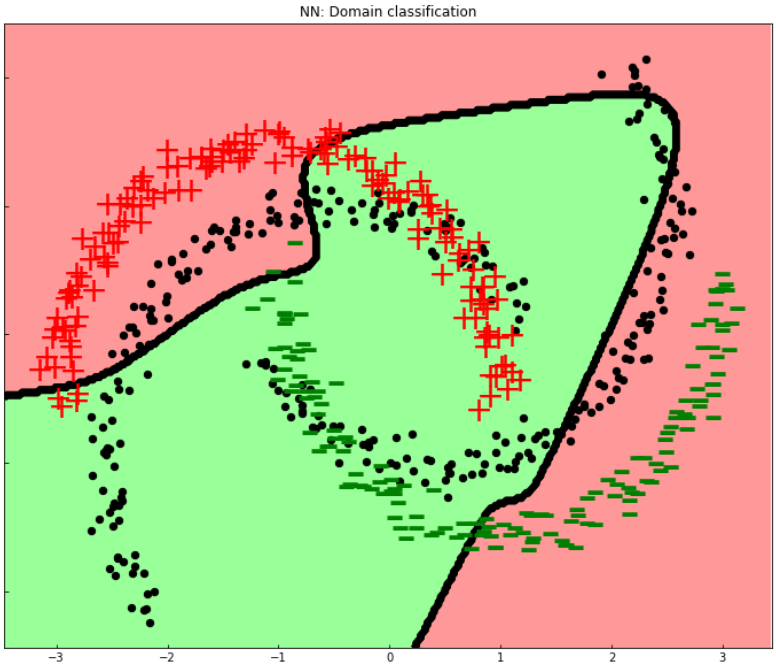
NN: Label Classification :



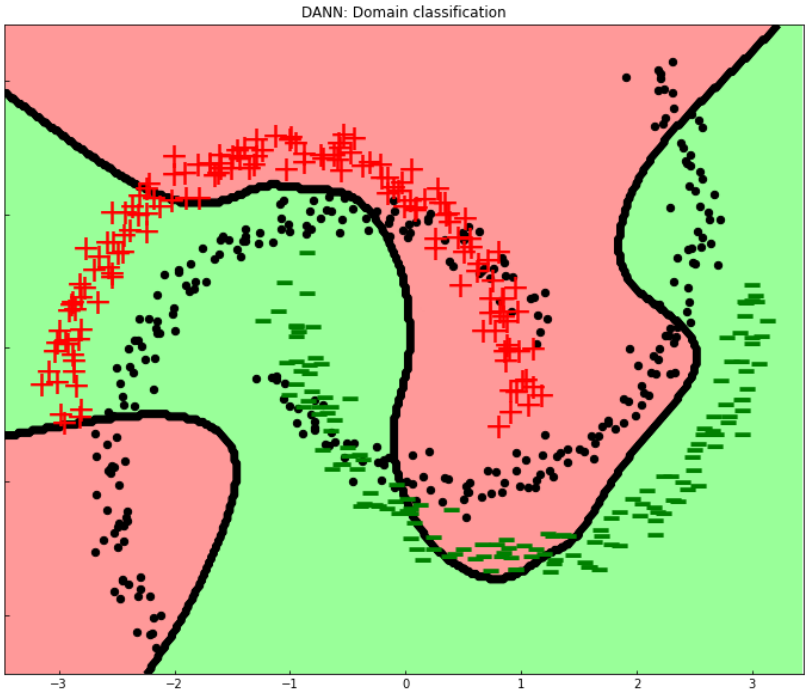
DANN: Label Classification :



NN: Domain Classification



DANN: Domain Classification



Observations:
DANN is more domain invariant and label discriminant.
The decision boundary for label classifcation is better in case of DANN.
The decision boundary for domain classifcation is better in case of NN.

3. Results

NN :

- Label classification accuracy of the source test set: 1.000000
- Domain classification accuracy of the source test set: 0.579167
- Label classification accuracy of the target test set: 0.695833
- Domain classification accuracy of the target test set: 0.595833

DANN :

- Label classification accuracy of the source test set: 1.000000
- Domain classification accuracy of the source test set: 0.441667
- Label classification accuracy of the target test set: 0.983333
- Domain classification accuracy of the target test set: 0.466667

Accuracies in tabular form on

Model	(Source) Label	(Source) Domain	(Target) Label	(Target) Domain
NN	1.0	0.579	0.696	0.596
DANN	1.0	0.442	0.983	0.467

Google Drive links for the saved models:

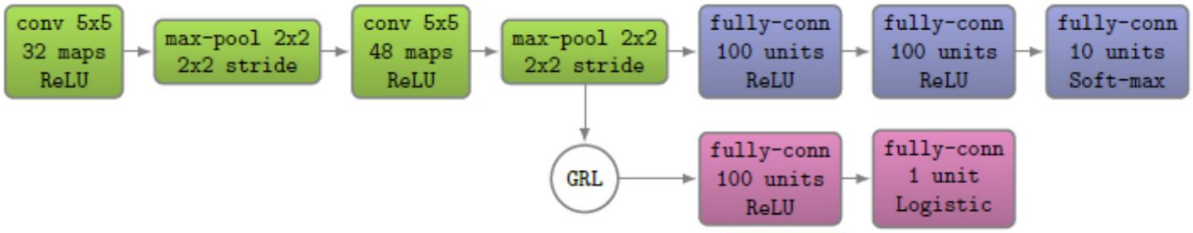
DANN: link

NN: link

Part 2)

1. Architecture:

Same architecture has been implemented as given in the paper(depicted below). The green cells constitute the feature generator, the purple cells constitute the label classifier and the pink cells constitute the domain classifier. For the source only model, only the green and purple boxes have been incorporated.



(a) MNIST architecture; inspired by the classical LeNet-5 (LeCun et al., 1998).

2. Data :

Source Domain : MNIST
Target Domain : MNIST-M

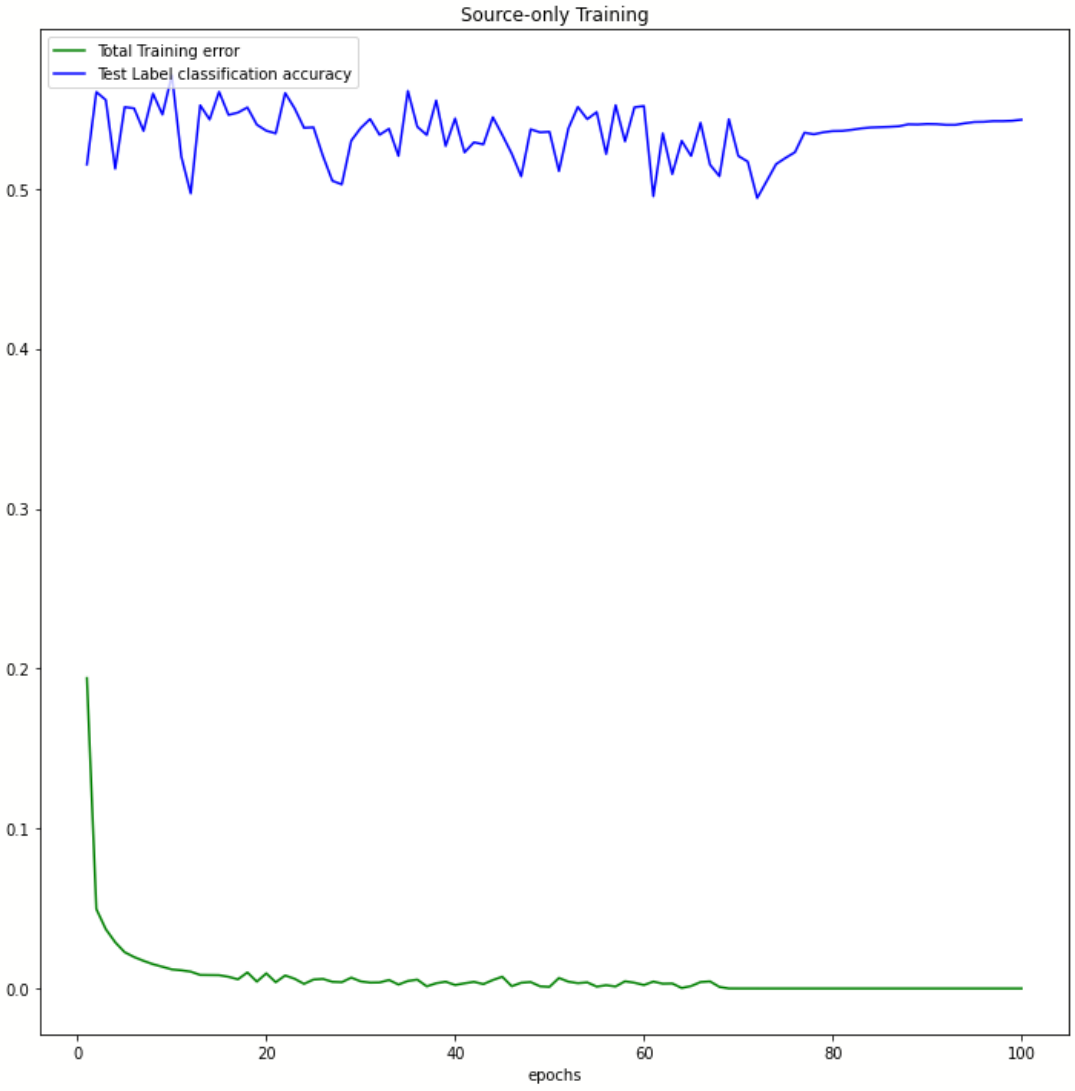
Preprocessing of data:

Preprocessing steps included resizing the image to 28x28 and normalizing the images. The inbuilt train-test splits were used for the datasets (85-15 approx).

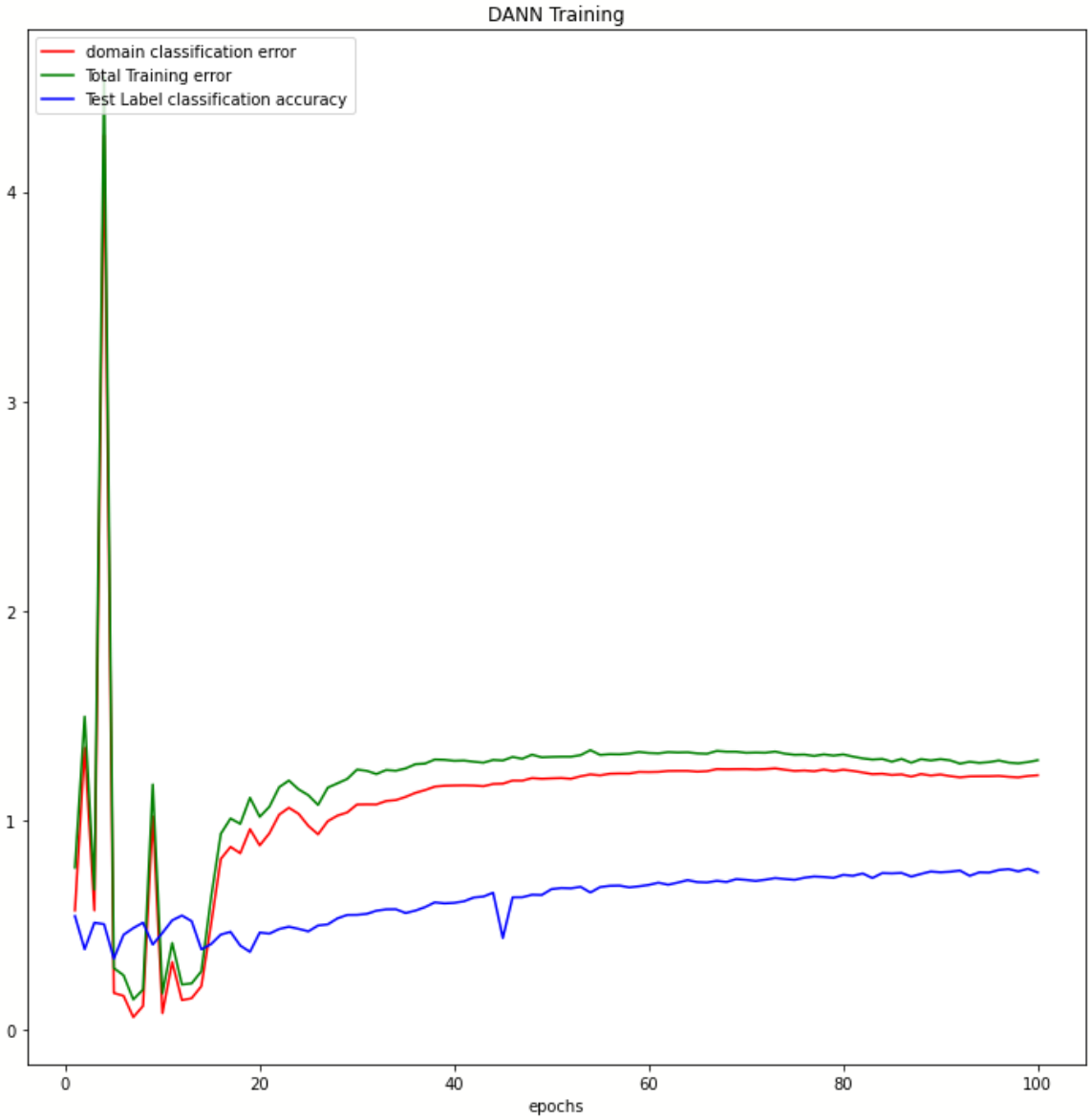
3. Training Details:

Learning Rate = 1e-3
batch_size = 128
image_size = 28
Optimizer = Adam
Number of epochs for Source_only model = 40
Number of epochs for DANN = 100

Source-only Training Plot:



DANN Training Plot:



4. Results:

The accuracies for label classification and domain classification for test-sets of source and target domains have been reported below.
An attempt was made to replicate the results mentioned in the paper.

Model	Label(Source)	Label(Target)	Domain (Source)	Domain (Target)
Source-only	0.990800	0.522720	-	-
DANN	0.961300	0.769470	0.676900	0.644595

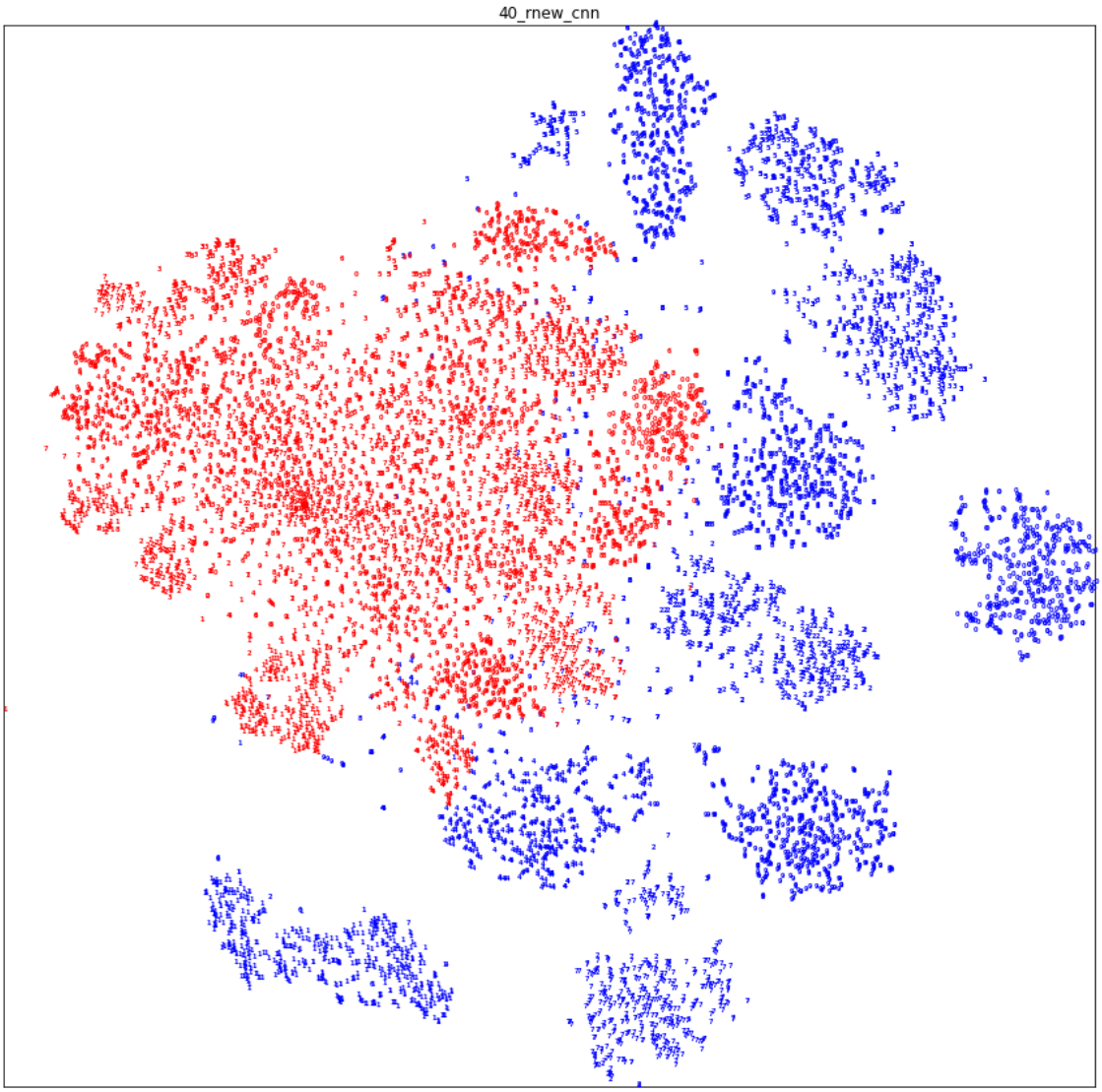
5. TSNE Plots:

The TSNE plots of the features generated for the data samples in test-sets of source and target domains have been plotted. The points in blue belong to the source domain and the points in red belong to the target domain.

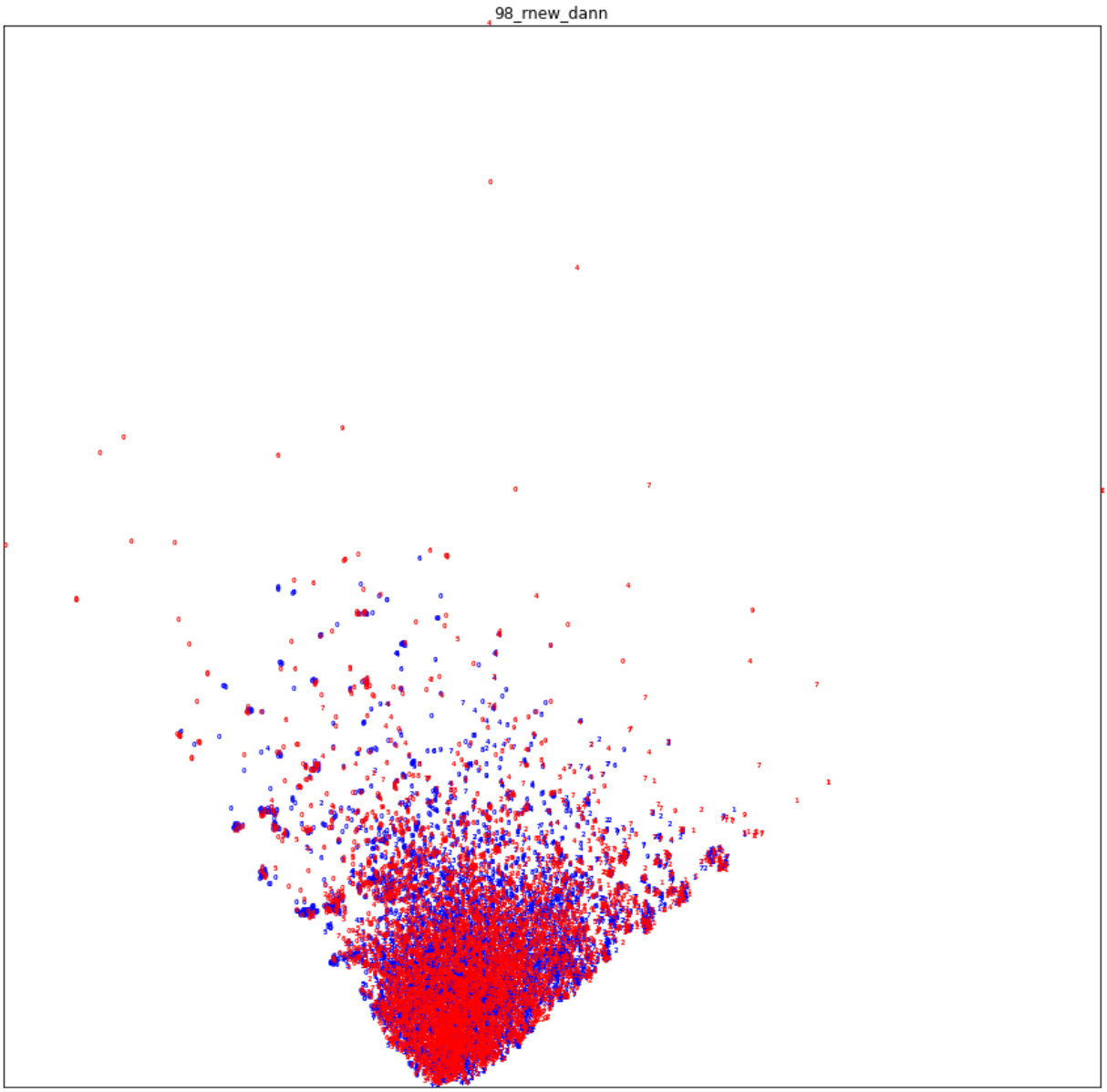
Observation: The TSNE representation produced by *source-only* model depicts points from source and target domain in spread-out clusters with very few overlaps highlighting that the distributions learnt by the model for the source and target domain are significantly different. However, the TSNE plot of *DANN* model depicts the two distributions to be very similar with a huge number of overlapping points. This emphasises the remarkably larger degree of domain invariance exhibited by *DANN* as compared to *source-only* model.

The domain invariance possessed by DANN helps it perform better label classification on the target domain as compared to Source-only while both models perform reasonably well on the source data.

TSNE Source-only Model:



TSNE DANN:



Links for the saved models:
Source_only
DANN