AML Assignment-5

-SUBHANI SHAIK(MT18117)

1. Scene Text Recognition using CRNN

Dataset: IIIT5K-word dataset

- The dataset has 2000 train samples and 3000 test samples.
- Each image is having 3 channels with variant sizes.
- Dataset has labels of the image and have small lexicons, large lexicons to that image.

Pre-processing data:

- Loaded the data and applied basic transformations like Resizing to 32,100 dimensions to all images, converting to grey-scale image and normalized using mean and standard deviation of 0.5.
- For labels taken characters from A-Z and 0-9 and encoded them by giving unique number and can also be decoded at test time.

Architecture and Procedure:

- Implemented the CRNN architecture which is a combination of CNN and LSTM components and last layer is of Transcription layer where we decode the output from fully connected layer and apply the loss function.

Type	Configurations
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	#maps:512, k:2 × 2, s:1, p:0
MaxPooling	Window:1 × 2, s:2
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
MaxPooling	Window:1 × 2, s:2
Convolution	#maps:256, k:3 × 3, s:1, p:1
Convolution	#maps:256, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:128, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:64, k:3 × 3, s:1, p:1
Input	$W \times 32$ gray-scale image

- Used Bidirectional LSTM and used different sizes of pooling layers after convolutional layers.
- For loss function used CTCLoss (The Connectionist Temporal Classification loss) which is used to calculate loss between continuous time series and a target sequence.

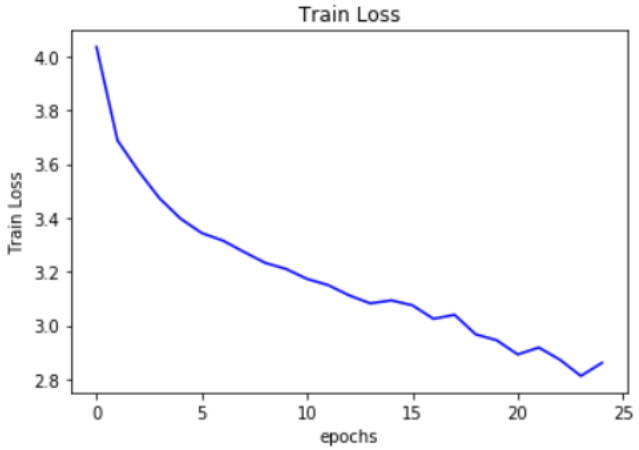
- Used the Adadelta optimizer for optimization for automatic calculate per-dimension learning rates.
- Taken batch size of 10 and trained the architecture for 50 epochs.

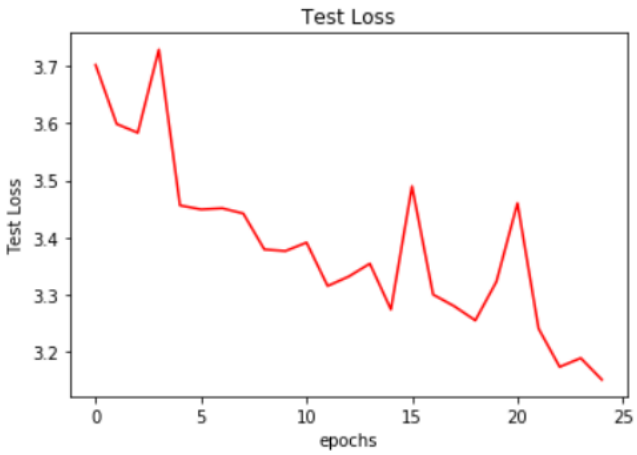
Analysis and results:

- Trained the network by this architecture and plotted the results of training and testing loss.
- As the network is progress in training the training loss and test loss both are gradually converging.
- After 25 epochs the train loss is still decreasing but test loss is fluctuating.
- Taken the test accuracies also, but as our predicted label is not exactly as same as actual label, it will give very less accuracy.
- After training tested on test dataset and plotted the result with their original labels.
- It was predicting approximately, some mistakes for some characters. But if I train more, then the model is overfitting.

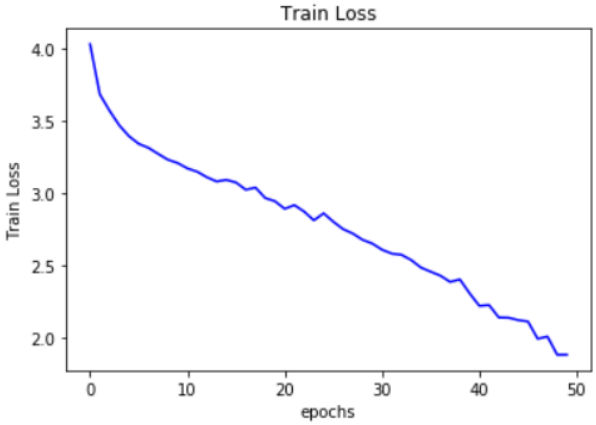
  [For information please see notebook]

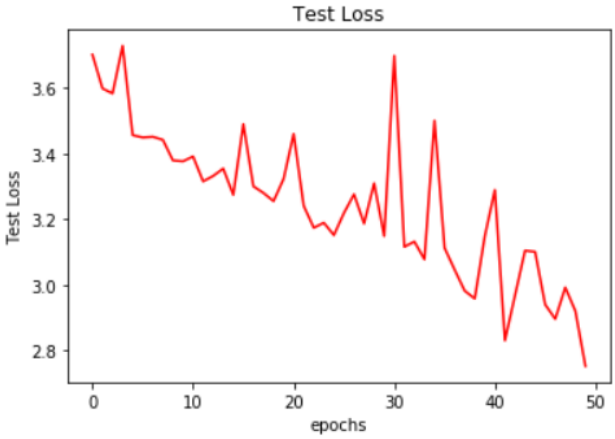
For first 25 epochs:





For first 50 epochs:





2. Image style transfer using GAN

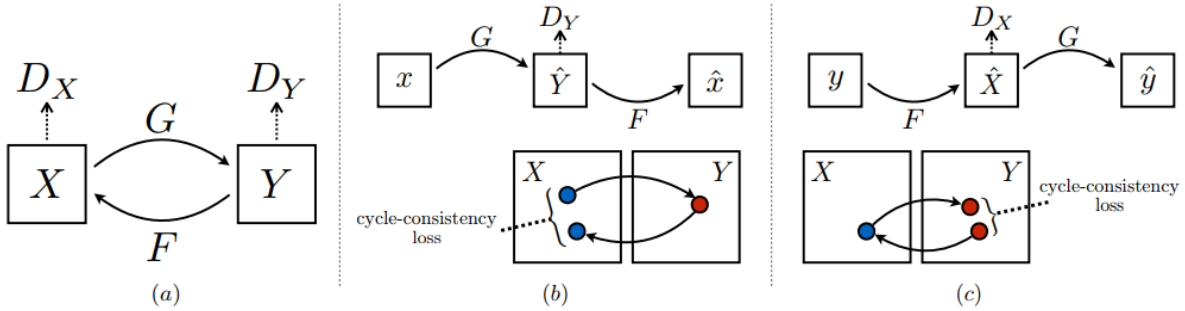
Dataset: Horse2Zebra dataset

- Taken Horse2Zebra dataset which has training set- 1067 points of Horse and 1334 points of zebra.
- And test set has 120 points of Horse and 140 points of zebra.
- For pre-processing resized the image to 256x256 image and normalized the images.
- Due to computation convenience, I have trained on 100 points of each class.

Architecture and Procedure:

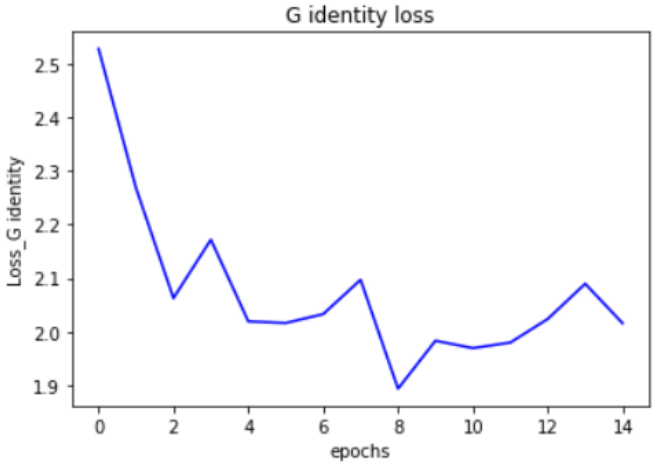
- I implemented the CycleGAN architecture which has two generators and two discriminators.
- It is a unpaired image translation from one class to another class.
- It has the cycle consistency loss.
- It uses Forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ *and* backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$.
- The network has generator and discriminator. Used Residual network for generator along with instance normalization and Leaky relu activation function.For Discriminator used normal CNN based architecture as classifier.

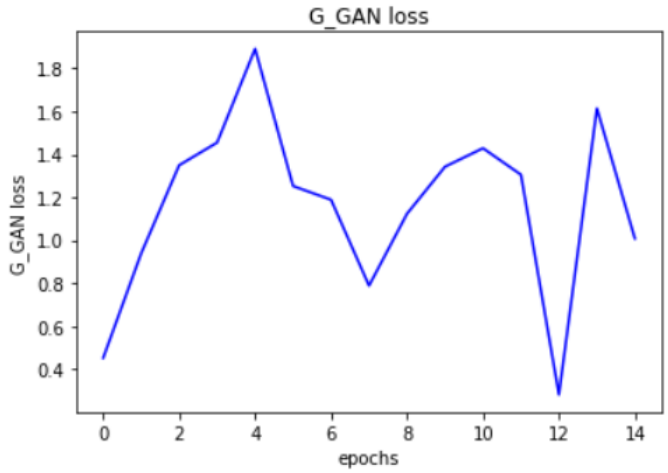
- Used the MSEloss for , L1 loss for cycle consistency.
- Trained the network using batch size of 1 and learning rate of 0.0002 and used Adam optimizer.

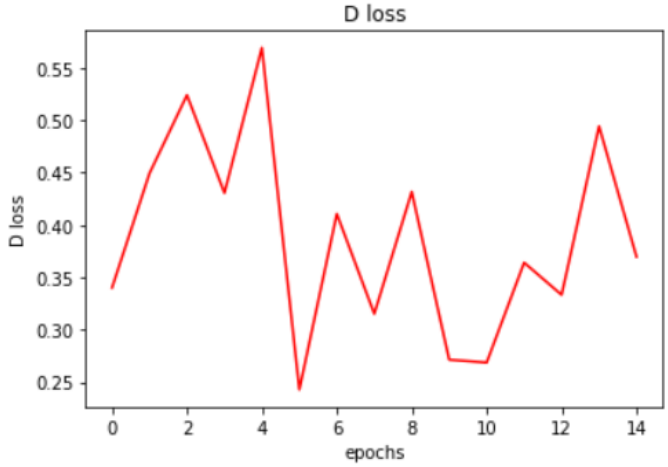


Analysis and results:

- I trained the network using Adam optimizer.
- Due to computation expensiveness, trained on sampled data points and plotted the results.







Output: Trained the network for 20 epochs and taken results. By training more we are getting better results.