Name: Punit K Jha
NetID:punit2
Team name on Kaggle leaderboard: PunitKJha

Part 1:

Answer the following questions briefly (no more than a few sentences), and provide output images where requested.

Show final results from training both your GAN and LSGAN (give the final 4x4 grid of images for both):

For training LSGAN and GAN I used the default hyperparameter setting shown below:

NOISE_DIM = 100
NUM_EPOCHS = 18 or 23
learning_rate = 0.0002

I ran GAN for 18 epochs (as per the advice from one of the TAs and also, I was running out of Google credits). The final results for this is shown below (with spectral decomposition).

Iter: 17700, D: 0.09034, G:4.887

I ran LSGAN for 23 epochs and got the following output (spectral decomposition included(. The final results for this is shown below (with spectral decomposition).

```
Iter: 22000, D: 0.01697, G:0.4569
```

Discuss any differences you observed in quality of output or behavior during training of the two GAN models.

For comparison purposes I have shown below output of GAN and LSGAN after 18 epochs. As is visible form the images below that LSGAN might be slightly better than GAN in terms of the overall quality of images produced. We can see that LSGAN images have are brighter and more colorful than the GAN images. In effect the LSGAN training was slower but more stable in terms of the generator and discriminator errors observed during training.

GAN Output

```
Iter: 17700, D: 0.09034, G:4.887
```

LSGAN Output

```
Iter: 17600, D: 0.01545, G:0.4
```

Do you notice any instances of mode collapse in your GAN training (especially early in training)? Show some instances of mode collapse (if any) from your training output.

Even after looking through the training outputs of GAN for 18 epochs and LSGAN for 23 epochs, I was unable to find an instance of mode collapse. However, I did find evidence of instability where even after LSGAN training for 20 epochs, I got the output images shown below. However, after a few iterations the instability disappeared, and I was getting good quality facial images again. I did not see such instability in GAN but that might be because I did not train it for more than 18 epochs.

Epoch No. 20

`Iter: 19400, D: 0.004181, G:0.4692`

Epoch No. 20

`Iter: 20200, D: 0.005627, G:0.5813`

Discuss briefly how/whether spectral normalization helps generate higher quality images in your implementation. Ideally, you should show samples from models with and without normalization.

As per the spectral normalization paper (https://arxiv.org/pdf/1802.05957.pdf) one round of power iteration was enough so I did not loop over the power iteration variable. This choice is justified by the results obtained which did not show any abnormalities.

Yes, spectral normalization (SN) does help in generating higher quality image in my implementation. Shown below are two images obtained using GAN after 11 epochs (beginning of the 12th epoch) with and without spectral normalization. As you can see the image produced by GAN implementation with SN is a remarkable improvement over the GAN without SN.

GAN with SN after 11 epochs

GAN without SN after 11 epochs

The difference was still prominent even after 18 epochs. Shown below are the image from GAN with and without spectral normalization after 18 epochs.

GAN with SN after 18 epochs

GAN without SN after 18 epochs

<u>Extra credit:</u> If you completed the extra credit for this portion, explain what you did (describing all model changes and hyperparameter settings) and provide output images.

I tried to change the model to get the full resolution 128x128 but it was very difficult and I kept running into errors after modifying the model to accommodate it to the new dimensions. Even trying to modify the model for the 64x64 images required a lot of debugging (especially the layer dimensions after every convolution layer) and in the end the results that I got were not satisfactory. Hence, using the same model I carried an exhaustive study of changing the hyperparameters provided to us in the notebook whose results are presented below.

Learning rate: Due to computational and time constraints I could train and compare the difference in image quality only of 5 epochs for different learning rates. Shown below are the images obtained by using different learning rate on the GAN model (with spectral normalization). We see that given such less number of epochs, only 5, a learning rate of 0.001 produces the best results compared to others.

GAN with LR=0.0002

GAN with LR=0.001

GAN with LR=0.01

Number of Epochs- Although experiment with this hyperparameter may sound trivial however given the instability of the GAN architecture its important to study whether the image quality increases proportionally with increase in number of epochs or are there any unstable training regions in between. Shown below are images from LSGAN (with spectral norm) training at different intervals during the training process – at 5, 10,15, 20 and 23 epochs. As you can see from the images below that there is gradual improvement of image quality and accuracy as we increase the number of epochs but around 20 epochs there is a sudden crash in the quality of generator predictions which gradually recovers by the 23rd epoch. This shows the instability of the GAN method in general.

LSGAN after 5 epochs

LSGAN after 10 epochs

LSGAN after 15 epochs

LSGAN after 20 epochs

LSGAN after 23 epochs

Noise size: I also studied the effect of different noise sizes on the output images after 5 epochs. Shown below are the images obtained by using different noise sizes on the GAN model (with spectral normalization). For all the experiments with variable noise sizes a fixed learning rate of 0.001 was used. For GAN the latent hidden space, z, that it samples random points from is important. If latent space is small, the model will reach a point which it can't produce better quality anymore or will keep on producing the same images in all the 4x4 spots and if it's too big, the model might take a very long time to produce good results and sometimes as is the case of size=300, it may even fail to converge

Noise Size=50

Noise Size =100

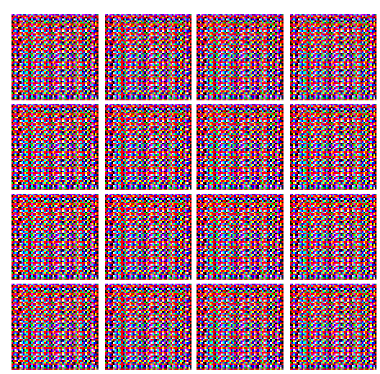
Noise Size =300

Adam optimizer: Another interesting aspect of the hyperparameters that were provided was that the Adam optimize has its exponential decay rate for the 1st moment estimates set to 0.500 instead of the default 0.999. So, as minor variation, I set the value back to 0.999 to see the effect. I also ran an experiment setting the 1st moment estimate to 0.300 In all the images from the experiments below a noise dim. of size 100, and learning rate of 0.0002 were used along with spectral normalization. As we can see that setting $\beta_1 = 0.999$ was not a good idea as the results did not converge. $\beta_1 = 0.30$ seemed to be like a slight improvement over $\beta_1 = 0.5$. So the Adam optimizer may not be as robust to the choice of hyper parameters as much as we think especially when it comes to GAN**.**

$\beta_1 = 0.300$ $\beta_2 = 0.999$

$\beta_1 = 0.500$ $\beta_2 = 0.999$

$\beta_1 = 0.999$ $\beta_2 = 0.999$



Part 2 (generation):

Give the hyperparameters for your best network on classification task below. Note any other changes you made to the base network in addition to the hyperparameters listed in the table below.

1. **List of different RNN types that I tried—**I implemented these models rnn, lstm, gru.
2. **List of the Number of layers that were tested— 1,2,3 –** Finally, given the trade off between computational time and accuracy, I chose to go with 1 layer which was faster and gave good enough test accuracy.
3. **List of hidden size layers tested—100,150,300,600 –** Due to the described trade-off above I choose 150 as the size of the hidden layers.
4. **List of learning rates tested—0.01,0.001,0.005**

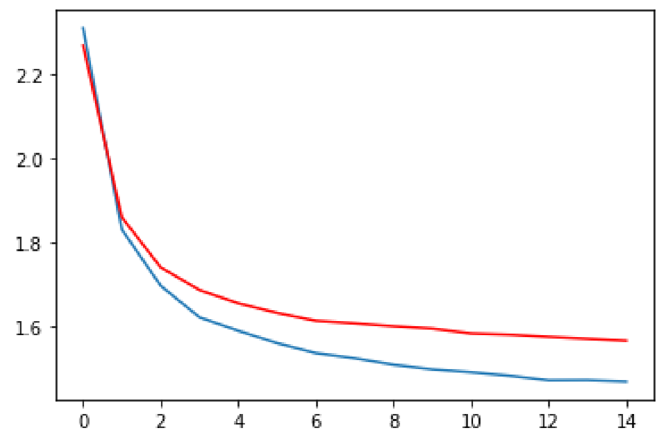
Hyperparameter	Value
RNN type:	lstm
Number of layers:	1
Hidden layer size:	150
Learning rate:	0.01

Give an example 1000 character output from your network:

```
The man wake him not hearing of his brother,
Even it but the great sorrow and have them: and
countrymen stands ill to seeming, the boy,
I misa man cave his heads: therefore hath restamed
thee my shadief in things and trust thou, for the faith
She is very eater in here.

ROSALINE:
Give this made it in the lander: you
see the day will necre 'gainst the heaps of heaven.

MARGARET:
I will take up the easy?

CARDINAL WOLSEY:
If your queen of the charess weep,
And supportune shook all the wait with him, he's
mine honest white high at all, which he small strength
```

```
them to hear a prier get descends and the good that,
With this place of my soul to secrets
And shall seen mine reputation, I'll answer he
made expled in our meaning strange plain.

HOLOFERNES:
Now, yet is not my husband on his admirally received
And must be how Perchain, is ever masters to them as
adversaries; which is to my sir, if you should
here a villain the better to be directed and confess of
the Frenchmen three suit; he is there
```

Insert the training & test loss plot from your RNN generation notebook below:



Extra credit: If you completed the extra credit for this portion, describe where your dataset came from, give an example from your training dataset (1000 characters), give an example output from your model trained on the dataset (1000 characters), and detail the hyperparameters you used to train a model on the dataset.

I downloaded 10 different novels of Charles Dickens from Project Gutenberg and concatenated them in one single file after some editing out certain portions of the file (removal of preface, references etc.)

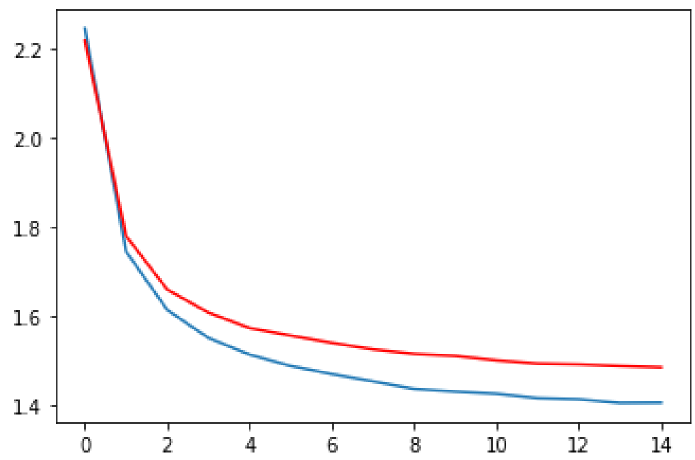
Here is a 1000 character example from my training dataset

My father's family name being Pirrip, and my Christian name Philip, my infant tongue could make of both names nothing longer or more explicit than Pip. So, I called myself Pip, and came to be called Pip.

I give Pirrip as my father's family name, on the authority of his tombstone and my sister,--Mrs. Joe Gargery, who married the blacksmith. As I never saw my father or my mother, and never saw any likeness of either of them (for their days were long before the days of photographs), my first fancies regarding what they were like were unreasonably derived from their tombstones. The shape of the letters on my father's, gave me an odd idea that he was a square, stout, dark man, with curly black hair. From the character and turn of the inscription, "Also Georgiana Wife of the Above," I drew a childish conclusion that my mother was freckled and sickly. To five little stone lozenges, each about a foot and a half long, which were arranged in a neat row beside their grave, and were sacred to the memory of five little brothers of mine,--who gave up trying to get a living, exceedingly early in that universal struggle,--I am indebted for a belief I religiously entertained that they had all been born on their backs with their hands in their trousers-pockets, and had never taken them out in this state of existence.

Since the LSTM run time was too high, I switched to gru from the extra credit portion since gru architecture was faster and the difference in accuracy was miniscule. The parameters and the plots are shown below.

Hyperparameter	Value
RNN type:	gru
Number of layers:	1
Hidden layer size:	200
Learning rate:	0.01



Extra credit: 1000 character output from my network trained on the works of Charles Dickens.

```
Is, and pulled to bear when he saw, and she was then or a point of Mr.
Sumble, as aspection, were no mading her handler stood for the stoods
pundsact.

"Day!"

"I want to you the most in you to see?" said Mr. Lorry. She had been offer
with mentions of lawburing his face. He had bad and with me to her hands to
be to set of thrify found impury for it what leading in her against suffed
people."

I saw, and paused it hurry, stood Defarge; "am this, she was the supporwaid
for good stand, to have you wish is a gently enlangeasure in the controod so
he proceeding of the Clara, and the piers, where me to be twomouson was never
appeared
to have a beginner, as the likely to the indived into the table instate at
her feel on the possible upon his astreesers (I want to the little touched to
her fast came and
graught," returned him of me," replied the word
with his last were pursuit. At the ainher. And me where she had relieved the
coversame, and the kitches and said Joe
was find a warms and wants repa
```

Part 2 (classification):

Give the hyperparameters for your best network on classification task below. Note any other changes you made to the base network in addition to the hyperparameters listed in the table below.

1. **List of different RNN types that I tried— rnn, lstm, gru** – I finally chose to use LSTM as I had a target of 90 percent on the Kaggle competition.
2. **List of the Number of layers that were tested— 1,2,3-** I chose 2 layers as it was a good balance between computational speed and accuracy.
3. **List of hidden size layers tested—100,300,600,1000 –** 600 was the optimal choice. Hidden size of 1000 took too long to run.
4. **List of learning rates tested—0.01,0.001,0.005**

Hyperparameter	Value
RNN type:	lstm
Number of layers:	2
Hidden layer size:	600
Learning rate:	0.001

You should reach the Kaggle accuracy benchmark with your Kaggle submission. Your notebook evaluation results should be similar to your performance on Kaggle. Insert the confusion matrix image outputted from your best model, and report the corresponding accuracy:

The final test accuracy was found to be 0.903

The initial training (blue) and test(red) loss plot for the first 2000 iterations is presented in Figure 1(a). This achieved an accuracy of about 98 percent. Then the initial model and its parameters were saved and reloaded for another round of 2000 iterations. This was required because I kept running into "insufficient RAM to run this operation" errors.

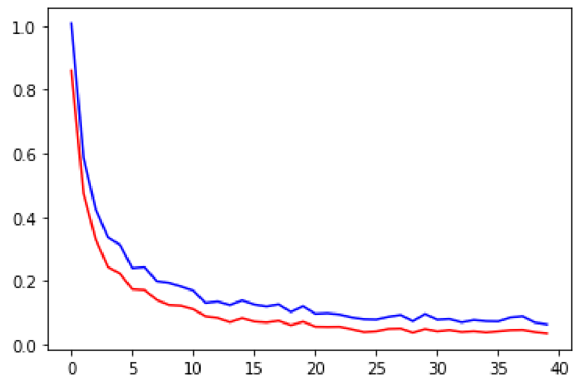


Figure 1(b)

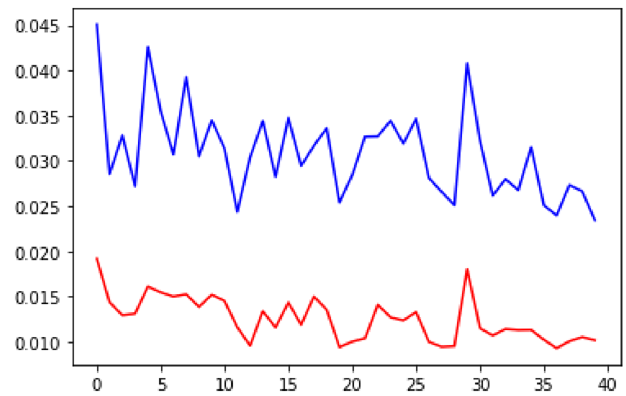


Figure 1(b)

Shown below is the confusion matrix image outputted from my best model. As we can see from the bright spots in the matrix that Xhosa, Romanian and Maori are the languages that our model learnt to predict the best.

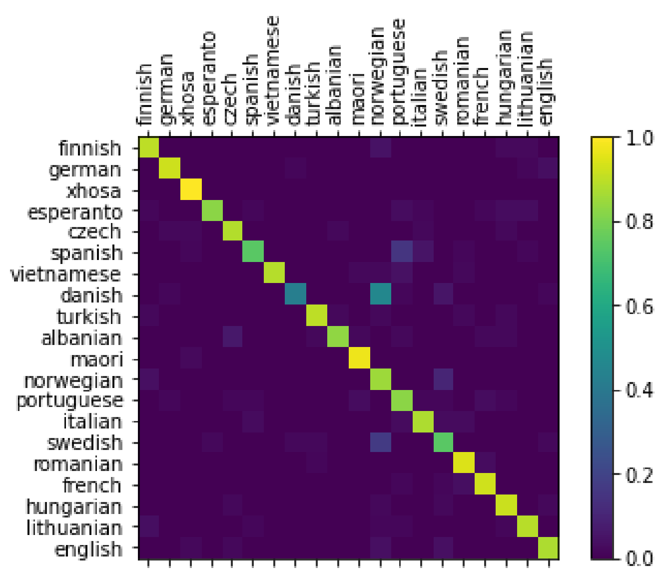


Figure 1