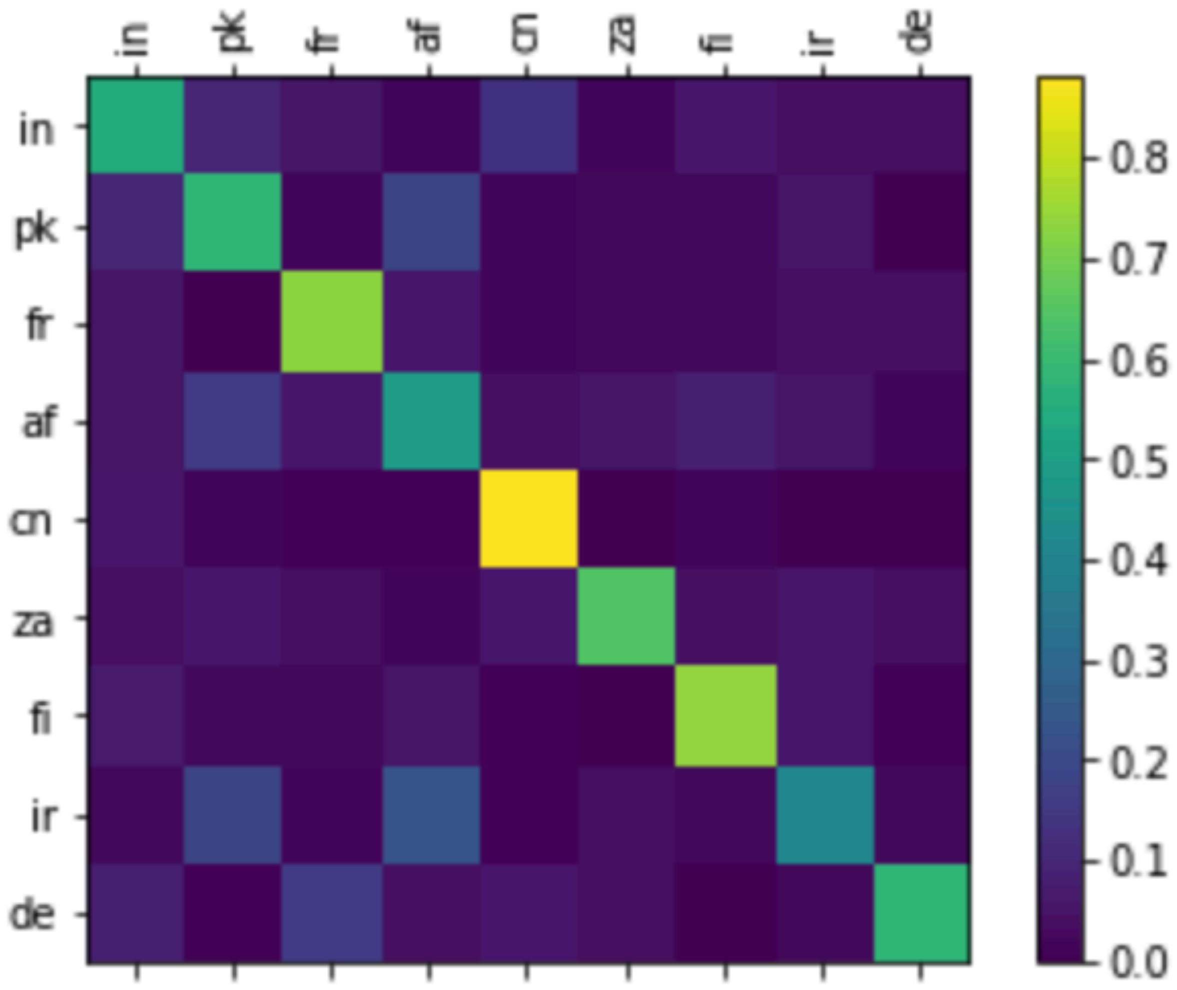
WriteUp-assignment3

U34502501 Haoran Wei

Part 1

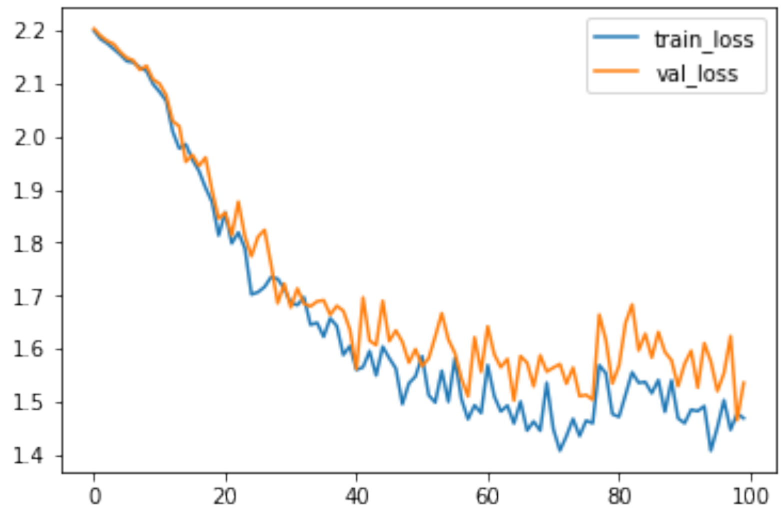
1. **Include your best validation accuracy in the report. Discuss where your model is making mistakes and use a confusion matrix plot to support your answer.**

```
macro Accuracy: 0.623333
micro Accuracy: 0.623333
```



Here in my model's accuracy on Validation dataset is around 0.6. In the picture, we can find that chinese can be detected pretty well. But for Afghanistan and Iran, they are confused by Pakistan and Afghanistan. Maybe there are some overlap between them because of their geolocations. Also, some language has number and english inside, which would be confusing if we want to detect them correctly.

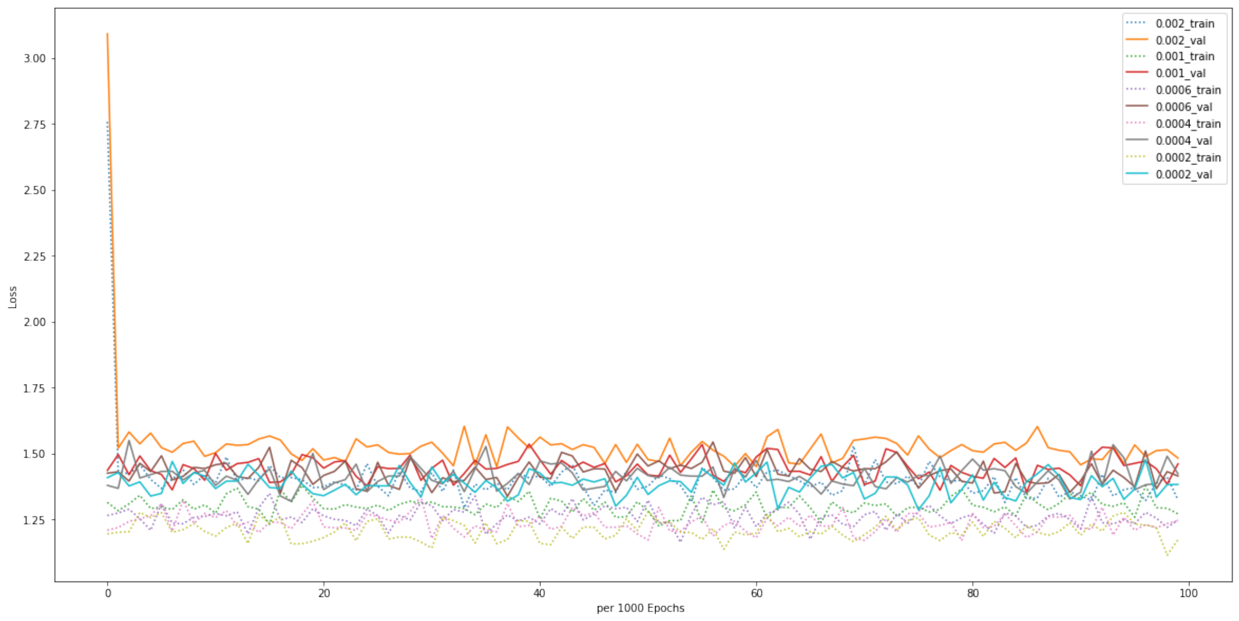
2. **Modify the training loop to periodically compute the loss on the validation set, and create a plot with the training and validation loss as training progresses. Is your model overfitting?**



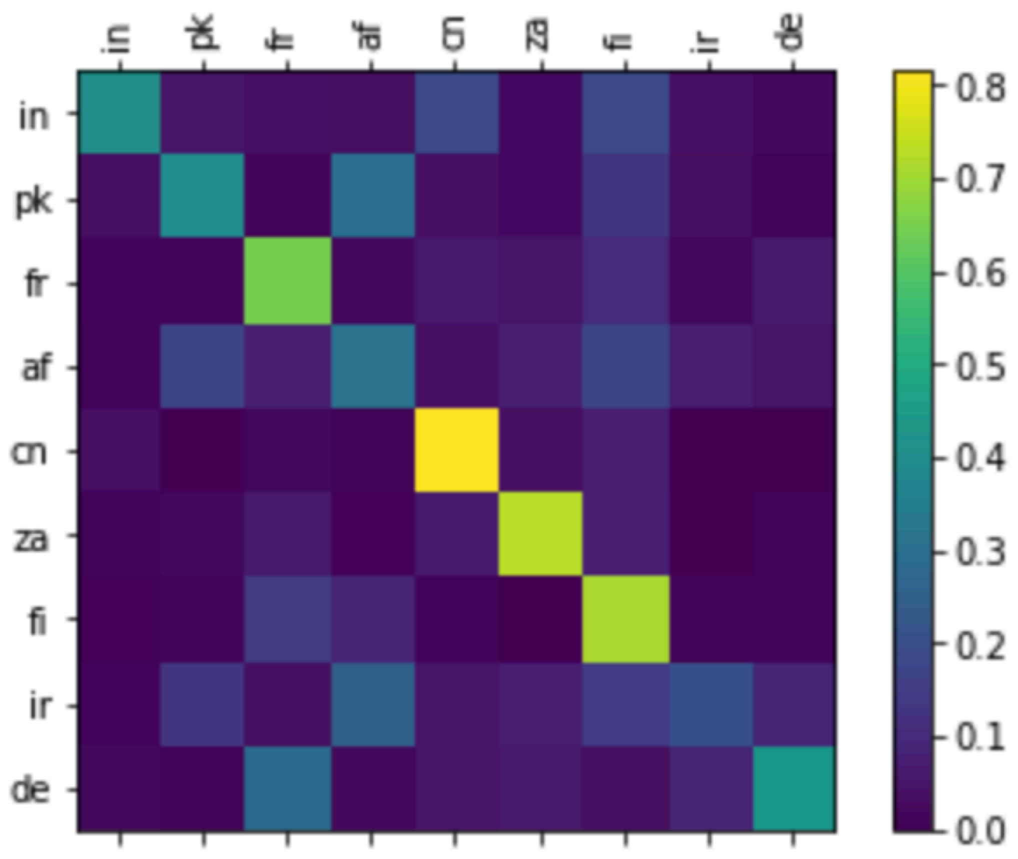
According to our loss curves with respect to train and validation, we could say that it begins as what we want, and then it becomes over 60 epochs, fluctuations happen. There are some overfitting after 80 epochs, as the train_loss goes down, val_loss goes up and down. However, we cannot clearly say that it is overfitting since val_loss still has a downside momentum.

3. **Experiment with the learning rate (at least 5 different learning rates). You can try a few different learning rates and observe how this affects the loss. Use plots to explain your experiments and their effects on the loss.**

   Here, I used 5 different learning rate in a decreasing fashion. As the learning rate descent, the train_loss could slightly increase and val_loss could slightly decrease comparing to larger learning rate.

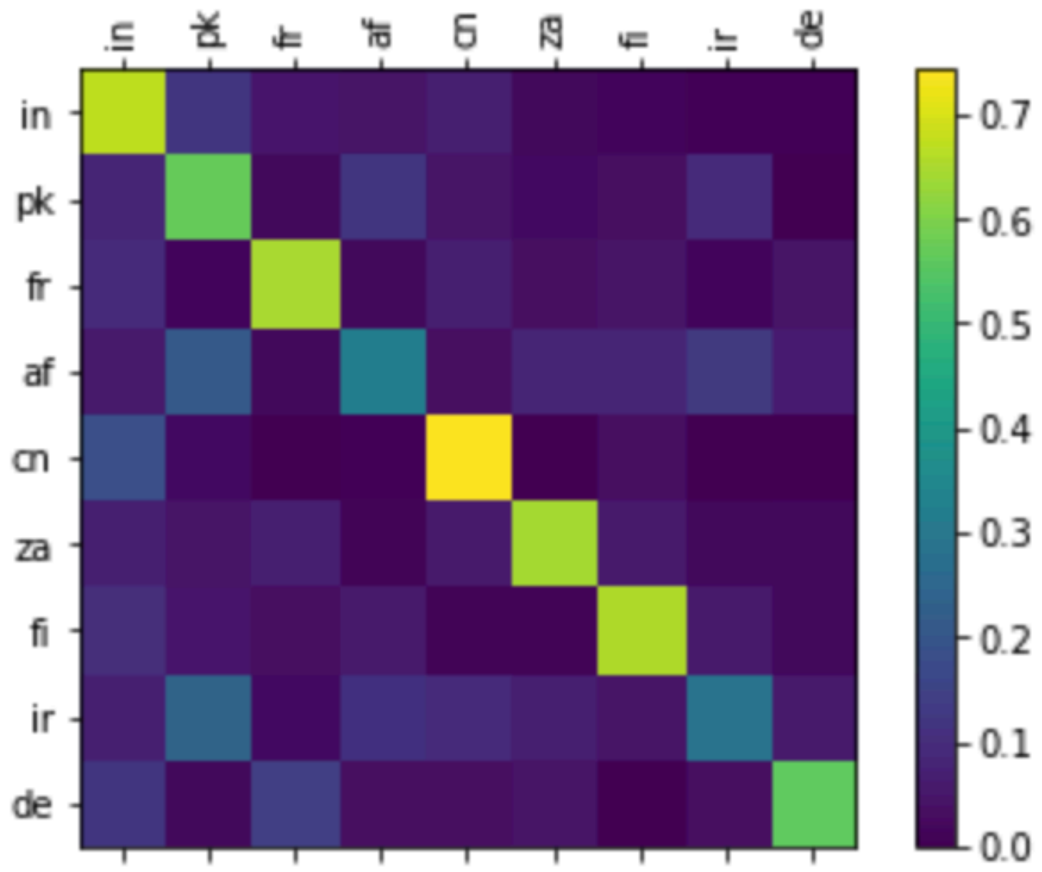


4. **Experiment with the size of the hidden layer or the model architecture (at least 5 different sizes and/or modifications). How does this affect validation accuracy?**
    1. *add one more layer for rnn. The result seems not much improvement.*
    2. *add activation function relu and tanh for the first layer output. The result for relu seems promising, the accuracy goes from 0.48 up to 0.51*



```
macro Accuracy: 0.519495
micro Accuracy: 0.521700
```

3. *change the optimizer in RNN, the result improves much from 0.51 to 0.56*



```
macro Accuracy: 0.568488
micro Accuracy: 0.566600
```

4. change the learning rate from 0.002 to 0.0002, the result reaches up to 0.62.
5. Reduce the number of hidden layer nodes, not much improvement or even worse.

```
macro Accuracy: 0.623333
micro Accuracy: 0.623333
```

Some result is shown in the improvement directory.

Part 2

Here I use different dataset from nottingham music. Shown in the music folder. If you are looking for full results, please look up assignPart2.ipynb

Q1:Include a sample of the text generated by your model, and give a qualitative discussion of the results. Where does it do well and where does it seem to fail?

```
'T:Aly Missin\n% Nottingham Music Database\nS:Trad, arr Phil
Rowe\nM:6/8\nK:G\nP:A\nB/2|"G"GG "G"GG/2G/2|"G"G'
```

I trained my rnn based on .abc format music notations from nottingham music. The generated text in the train part gave us a clear mind for the process of learning. First, it starts with repetitive letters, and then it varies. Some combinations came out after 1000 epochs. And it can, to some extent, form a correct format for abc. It was able to print the first severals lines with capital letters.

There are also some defects for them :

```
'T:Arest 1985, via EF\nM:4/4\nL:1/4\nK:D\n"D"AF "A7"AB|"A"A2 A2|"D"d2
d2:|\n\n\nX: 11\nT:The Eerst Woane\n% Nott'
```

Because nottingham music database is a noisy term. Some text generation cannot get rid of it or put them in the seperate area.

```
T:Arest 1985, via EF\nM:4/4\nL:1/4\nK:D\n"D"AF "A7"AB|"A"A2 A2|"D"d2
d2:|\n\n\nX: 11\nT:The Eerst Woane\n% Not
```

Some words even occurs in the music generation part which does not make sense.

Q2:Report perplexity on a couple validation texts that are similar and different to the training data.

I chose 3 type of test dataset.

The first one is xmas from nottingham music but this part isn't included in our training dataset. The perplexity for this is 1.25628.

The second is the all-abcs(provided by our professor). Unfortunately, I don't know which source it is from. But I think it is different from nottingham music dataset. We got 4.1302 for this one. Induced from the perplexity, we can say it is different from nottingham style of music.

The third is part of training file. So it would have low perplexity. We got 1.377 here. I am pretty confused of it, because it has worse perplexity than xmas dataset which was not trained.