Programming Assignment 2

Liam Miller, 101080769
Tarek sanger, 101059696

Question 1

For the 5 different CNN models we we tried different variations of the number of Convolutions adn MaxPool Sizes. All tests had a validation accuracy ranging from 65-75%.

What did we try?

In the first model we sa y a bit of over fitting which we tried solving with the L2 norm kernel regularizer - This did seem to solve the problems.

For the Convolutions we tried increasing their number as well as increasing the number of filters.

For the MaxPooling we kept to pool sizes of 2x2 and 3x3 and found that the combination of both gave the best results. Where we had a series of convolutions followed by a max pooling with size 3 and again a series of convolutions followed by a max polling with pool size 2.

Question 2

2.2

Description & Final Model

Tried doubling and halving the input size and the learning rate.
Substantial differences in validation accuracy.

- Hidden layers neurons of 256, 128, 64
- Increased and decreased the learning rate

Table of Validation Results

Model	Hidden Layer Size	Learning Rate	Validation Results
1	128	0.002	47.76%
2	128	0.0015	48.71%
3	128	0.001	48.59%
4	64	0.002	48.00%
5	256	0.002	51.76%

The best model ended up being our final model, Model 5, which had a learning rate of 0.002 and a

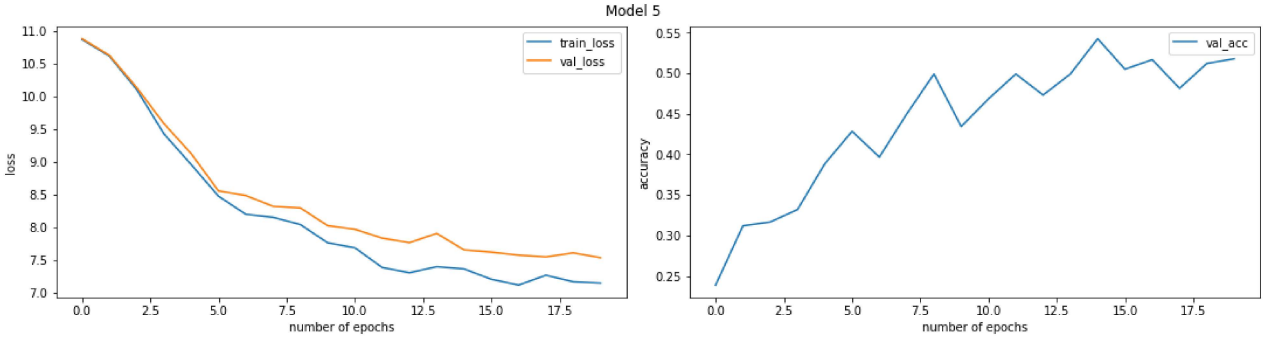
Hidden Layer Size of 256.

Best Model Plots (Model 5)

In [ ]:
```
'''
    We need to save the model validation accuracy so we can plot it here.
'''
import os
from IPython.display import Image

Image('accuracy_plot.png')
```

Out[ ]:

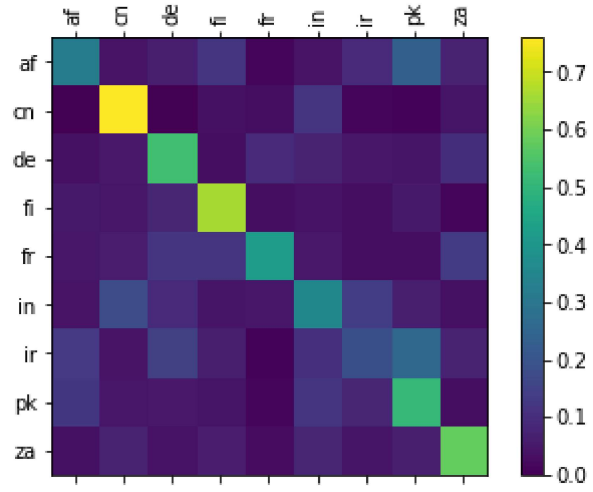


Confusion Matrix

In [ ]:
```
'''
    Display Confusion Matrix Here
'''

Image('confusion.png')
```

Out[ ]:



Error Analysis:

Struggles to properly classify "ir", confusing it with pk, and de often.
Does well with most other 2 letter combinations although it does mistake pk and af a lot as well.

2.3

For part 2.3 we used the Jane Austen novels. This text dataset actually an eBook of the complete works of Jane Austen, by Jane Austen.

The model being run is a text generation model that trains using random chunks of text in order to learn what letters will proceed two letter pairs. The hidden layer size is 100 and the learning rate is 0.005.

Example Generation

Input	Output
The	Ther, for had and nothing after whibest manness even a the provence; and "nower herser a gratia to her but a givemently to supe; and the usuned I shoove such the provighment not to he my disualy a good h
West	Westrestion; and the strution of on for in the--the out can her see coldlyn her were of her improve!" her. The dear such a greathing sillimen to her must aymarne."
Penguin	Pisencess could her were to this beas Penguing a looking a fermulal prose simpinecting in has and sure, and sher understincs, and and very that had gentlead comement of has not a good had her landens, I am a her from I aw band Fanny's with but a
Cake	Cakention, and hers object, by the him." I am consuable an her while jouse so life her for his be the much of as I had like a good him soon for convenfing jood have such aterable a s the paling though a

Qualitative Discussion of Results

The generator seems to be proficient at generating small connecting words in an attempt to construct a sentence. Many smaller words generated are real words. It seems to fail to produce accurate larger words. It also tends to place quotations where they are unnecessary. It also sometimes makes spaces in an attempt to start new paragraphs even though it is mostly unnecessary.

Perplexity

The average perplexity for a similar dataset was 5.89
The average perplexity for a very different dataset was 7.40