Task 2:

I have used the nps_chat corpus provided in NLTK. I have taken a sub corpus containing only 4 classes.

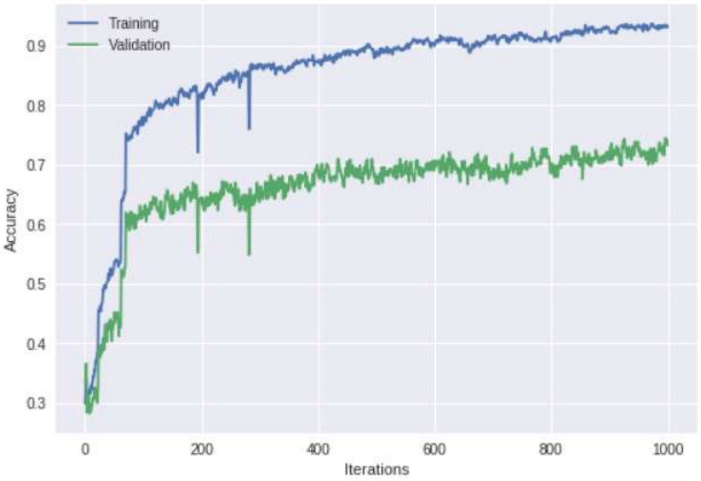
Classes	Train_Instances	Validation_Instances	Test_Instances
Greet	900	200	163
System	1000	200	132
Statement	1000	200	185
Emotion	800	200	106

I have used parser provided in spacy to tokenize the text and remove stopwords and punctuations. However, removing punctuation instead decreased the overall accuracy of the model.

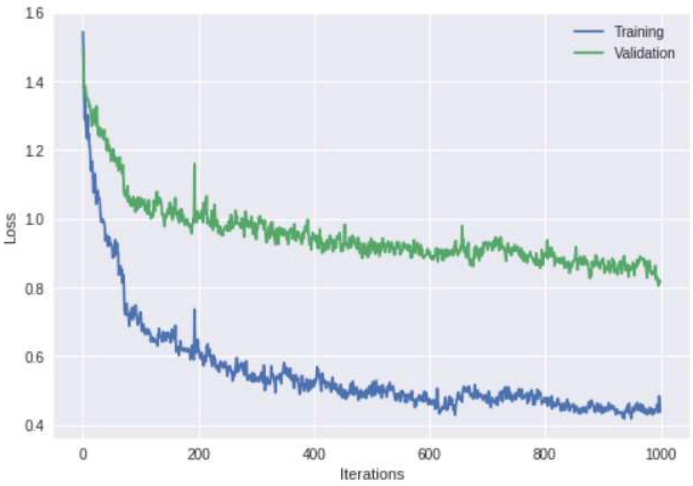
The texts has been vectorized using TfidfVectorizer() provided in sklearn module. It seemed easier to implement this. I have implemented a 4 layered feed forward network using tensorflow. Below are the outputs from the model:

```
Starting computation....
epoche 0 : train_loss = 1.54, train_acc = 0.2992, val_loss = 1.50, val_acc = 0.3350
epoche 100 : train_loss = 0.72, train_acc = 0.7805, val_loss = 1.05, val_acc = 0.6187
epoche 200 : train_loss = 0.62, train_acc = 0.8105, val_loss = 1.01, val_acc = 0.6350
epoche 300 : train_loss = 0.54, train_acc = 0.8654, val_loss = 0.94, val_acc = 0.6538
epoche 400 : train_loss = 0.51, train_acc = 0.8716, val_loss = 0.94, val_acc = 0.6812
epoche 500 : train_loss = 0.48, train_acc = 0.8886, val_loss = 0.93, val_acc = 0.6712
epoche 600 : train_loss = 0.46, train_acc = 0.9016, val_loss = 0.90, val_acc = 0.7075
epoche 700 : train_loss = 0.47, train_acc = 0.9151, val_loss = 0.91, val_acc = 0.6938
epoche 800 : train_loss = 0.48, train_acc = 0.9135, val_loss = 0.89, val_acc = 0.7025
epoche 900 : train_loss = 0.45, train_acc = 0.9243, val_loss = 0.88, val_acc = 0.7113
Training done!
Test_Accuracy: 0.7696793
```

```
<matplotlib.legend.Legend at 0x7fd8e3875550>
```



```
<matplotlib.legend.Legend at 0x7fd8e4e175c0>
```


Training
Validation
Loss
Iterations