Untitled0

September 3, 2019

[ ]:
```
```

[ ]:
```
from google.colab import drive
import os
import sys
drive.mount('/content/drive')
dep_path = os.getcwd()+'/drive/My Drive/outputs/'
sys.path.insert(0, dep_path)
os.listdir(dep_path)
```

[0]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import cm
import pandas as pd
```

1 Pre-Trained

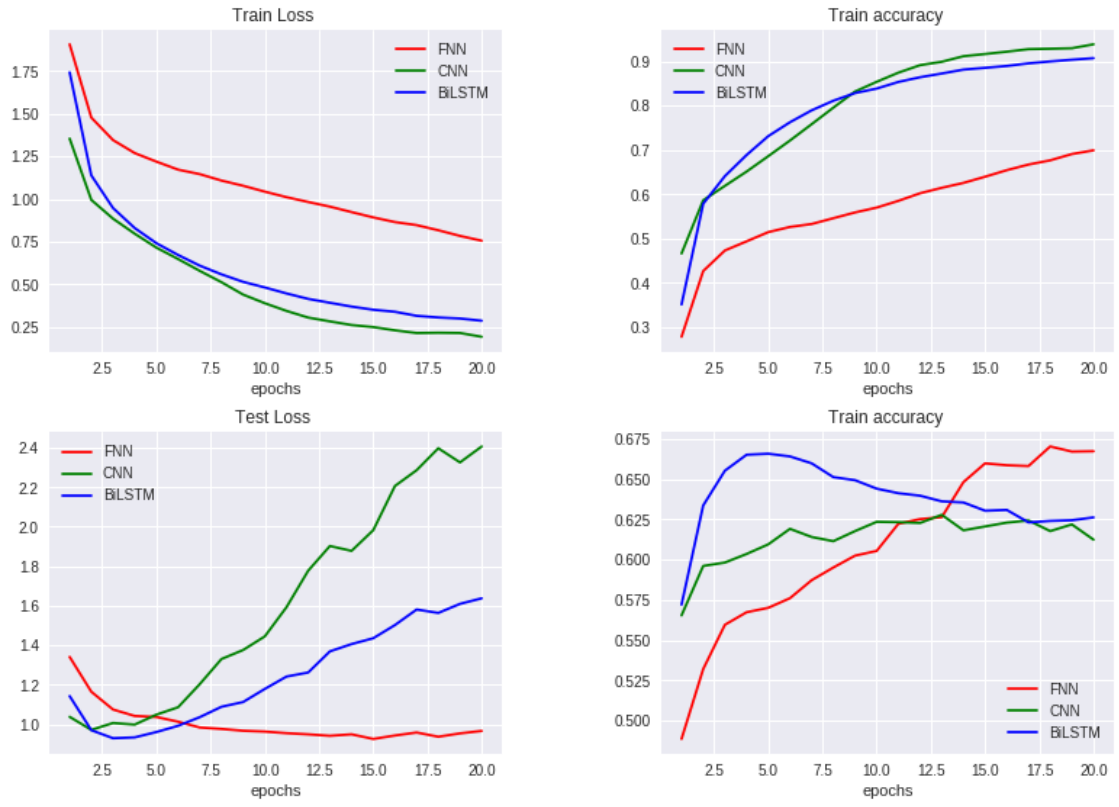
1.1 Customer Complaint

[0]:
```
pre_cust = np.load(dep_path+'pre_cust.npy').item()
pre_cust.keys()
```

[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)

y = list(range(1,len(pre_cust['CNN']['acc'])+1))
plt.title('Train Loss ')
plt.plot(y,pre_cust['NN']['loss'],color='r',label='FNN')
plt.plot(y,pre_cust['CNN']['loss'],color='g',label='CNN')
plt.plot(y,pre_cust['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(y,pre_cust['NN']['acc'],color='r',label='FNN')
plt.plot(y,pre_cust['CNN']['acc'],color='g',label='CNN')
```

```
plt.plot(y,pre_cust['BiLSTM']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(y,pre_cust['NN']['val_loss'],color='r',label='FNN')
plt.plot(y,pre_cust['CNN']['val_loss'],color='g',label='CNN')
plt.plot(y,pre_cust['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Train accuracy ')
plt.plot(y,pre_cust['NN']['val_acc'],color='r',label='FNN')
plt.plot(y,pre_cust['CNN']['val_acc'],color='g',label='CNN')
plt.plot(y,pre_cust['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```

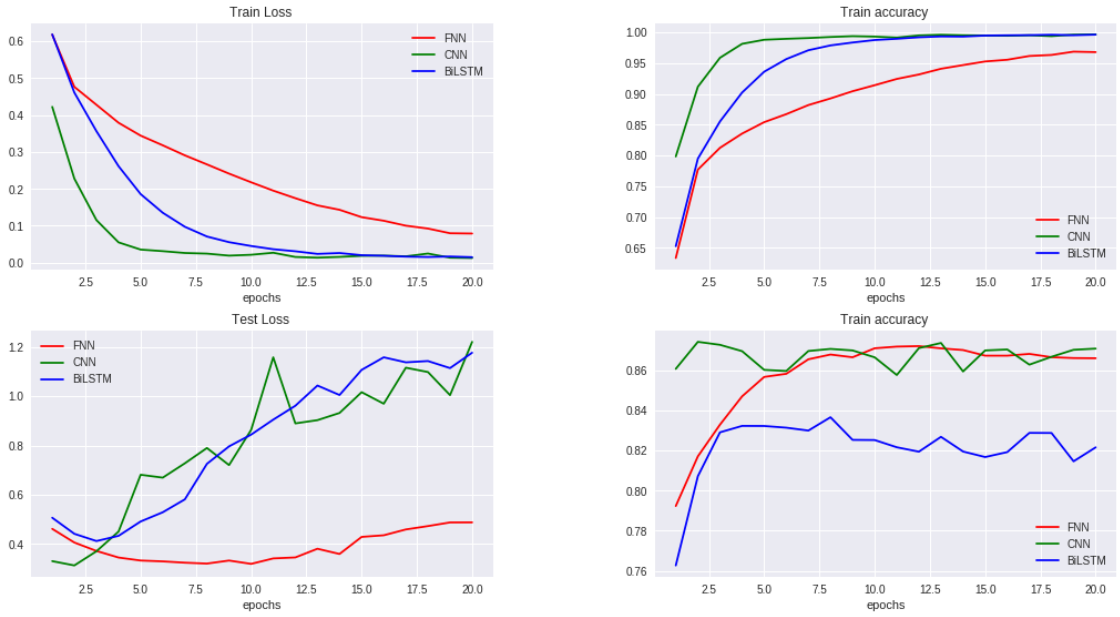


1.2 IMDB

[0]:
```
pre_imdb = np.load(dep_path+'pre_imdb.npy').item()
pre_imdb.keys()
```

[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
fig1 = plt.gcf()

y = list(range(1,len(pre_cust['CNN']['acc'])+1))
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(y,pre_imdb['NN']['loss'],color='r',label='FNN')
plt.plot(y,pre_imdb['CNN']['loss'],color='g',label='CNN')
plt.plot(y,pre_imdb['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(y,pre_imdb['NN']['acc'],color='r',label='FNN')
plt.plot(y,pre_imdb['CNN']['acc'],color='g',label='CNN')
plt.plot(y,pre_imdb['BiLSTM']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(y,pre_imdb['NN']['val_loss'],color='r',label='FNN')
plt.plot(y,pre_imdb['CNN']['val_loss'],color='g',label='CNN')
plt.plot(y,pre_imdb['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Train accuracy ')
plt.plot(y,pre_imdb['NN']['val_acc'],color='r',label='FNN')
plt.plot(y,pre_imdb['CNN']['val_acc'],color='g',label='CNN')
plt.plot(y,pre_imdb['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*2, DefaultSize[1]*1.5) )
```



2 CBOW

2.1 Customer Complaint

[0]:
```
cbow_cust = np.load(dep_path+'cbow_cust.npy').item()
```

[0]:
```
cbow_cust.keys()
```

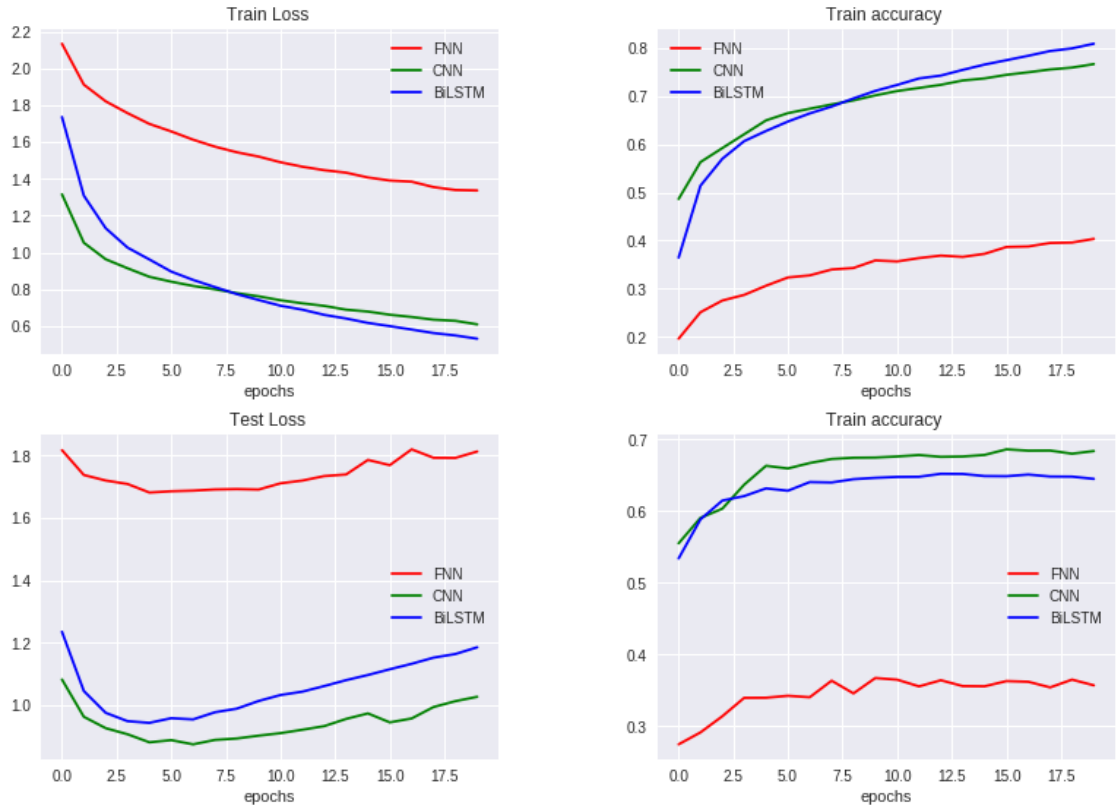
[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
cbow_cust['CNN'].keys()
```

[0]:
```
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(cbow_cust['NN']['loss'],color='r',label='FNN')
plt.plot(cbow_cust['CNN']['loss'],color='g',label='CNN')
plt.plot(cbow_cust['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(cbow_cust['NN']['acc'],color='r',label='FNN')
plt.plot(cbow_cust['CNN']['acc'],color='g',label='CNN')
plt.plot(cbow_cust['BiLSTM']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
```

```
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(cbow_cust['NN']['val_loss'],color='r',label='FNN')
plt.plot(cbow_cust['CNN']['val_loss'],color='g',label='CNN')
plt.plot(cbow_cust['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Train accuracy ')
plt.plot(cbow_cust['NN']['val_acc'],color='r',label='FNN')
plt.plot(cbow_cust['CNN']['val_acc'],color='g',label='CNN')
plt.plot(cbow_cust['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```



2.2 IMDB

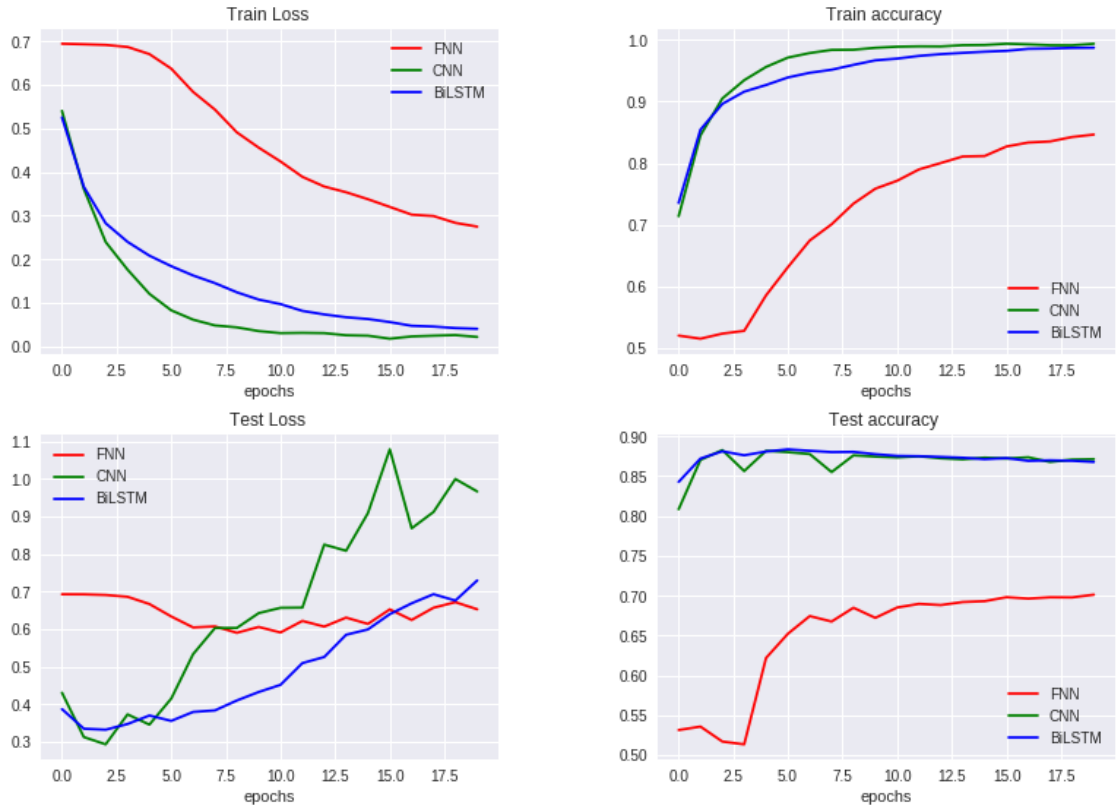
[0]:
```
cbow_imdb = np.load(dep_path+'cbow_imdb.npy').item()
cbow_imdb.keys()
```

[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
cbow_imdb['CNN'].keys()
```

[0]:
```
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(cbow_imdb['NN']['loss'],color='r',label='FNN')
plt.plot(cbow_imdb['CNN']['loss'],color='g',label='CNN')
plt.plot(cbow_imdb['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(cbow_imdb['NN']['acc'],color='r',label='FNN')
plt.plot(cbow_imdb['CNN']['acc'],color='g',label='CNN')
plt.plot(cbow_imdb['BiLSTM']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(cbow_imdb['NN']['val_loss'],color='r',label='FNN')
plt.plot(cbow_imdb['CNN']['val_loss'],color='g',label='CNN')
plt.plot(cbow_imdb['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Test accuracy ')
plt.plot(cbow_imdb['NN']['val_acc'],color='r',label='FNN')
plt.plot(cbow_imdb['CNN']['val_acc'],color='g',label='CNN')
plt.plot(cbow_imdb['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```



3 Skip-Gram

3.1 Customer complaint

[0]:
```
sg_cust = np.load(dep_path+'sg_cust.npy').item()
sg_cust['CNN'].keys()
```

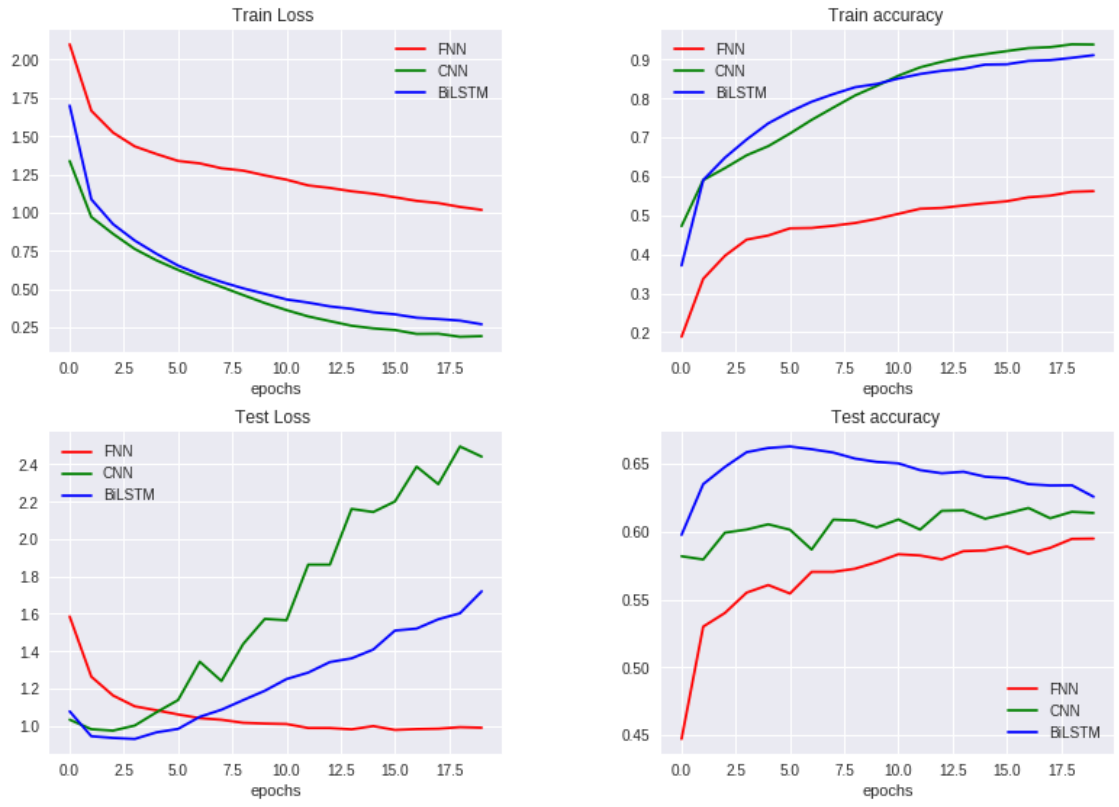
[0]:
```
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

[0]:
```
sg_cust['CNN'].keys()
```

[0]:
```
dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(sg_cust['NN']['loss'],color='r',label='FNN')
plt.plot(sg_cust['CNN']['loss'],color='g',label='CNN')
plt.plot(sg_cust['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
```

```
plt.plot(sg_cust['NN']['acc'],color='r',label='FNN')
plt.plot(sg_cust['CNN']['acc'],color='g',label='CNN')
plt.plot(sg_cust['BiLSTM']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(sg_cust['NN']['val_loss'],color='r',label='FNN')
plt.plot(sg_cust['CNN']['val_loss'],color='g',label='CNN')
plt.plot(sg_cust['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Test accuracy ')
plt.plot(sg_cust['NN']['val_acc'],color='r',label='FNN')
plt.plot(sg_cust['CNN']['val_acc'],color='g',label='CNN')
plt.plot(sg_cust['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```

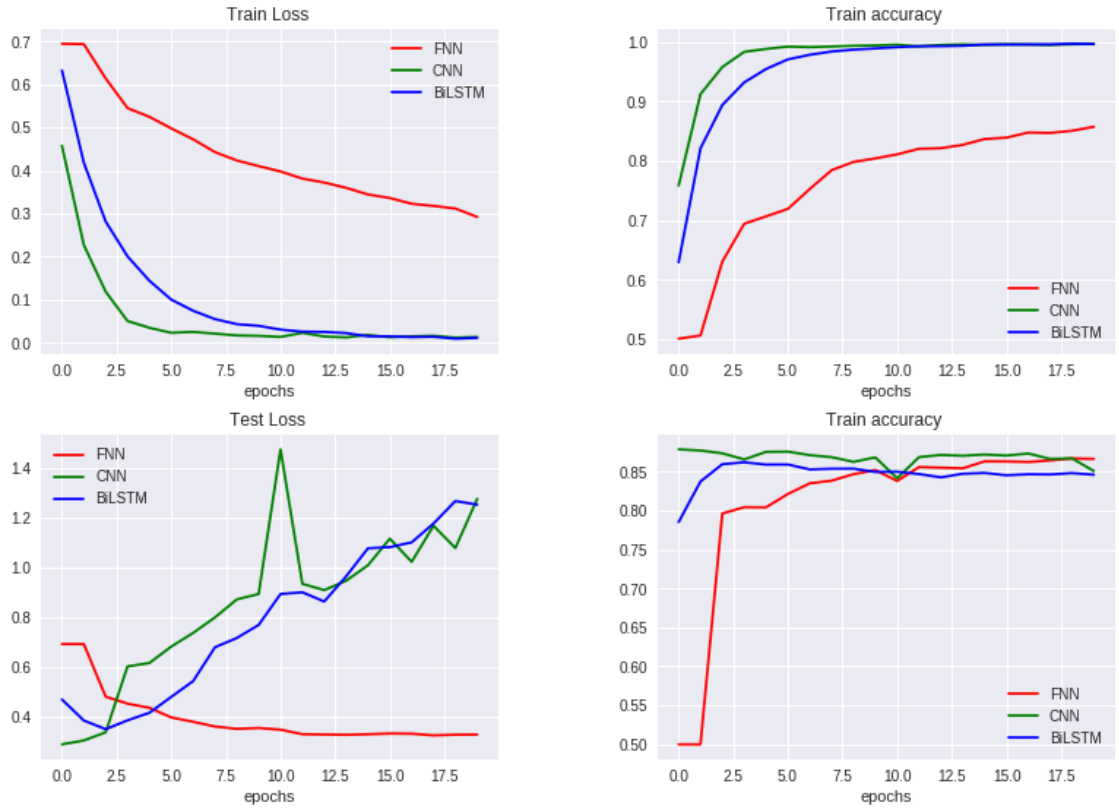


3.2 IMDB

[0]:
```
sg_imdb = np.load(dep_path+'sg_imdb.npy').item()
sg_imdb.keys()
```

[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(sg_imdb['NN']['loss'],color='r',label='FNN')
plt.plot(sg_imdb['CNN']['loss'],color='g',label='CNN')
plt.plot(sg_imdb['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(sg_imdb['NN']['acc'],color='r',label='FNN')
plt.plot(sg_imdb['CNN']['acc'],color='g',label='CNN')
plt.plot(sg_imdb['BiLSTM']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(sg_imdb['NN']['val_loss'],color='r',label='FNN')
plt.plot(sg_imdb['CNN']['val_loss'],color='g',label='CNN')
plt.plot(sg_imdb['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Train accuracy ')
plt.plot(sg_imdb['NN']['val_acc'],color='r',label='FNN')
plt.plot(sg_imdb['CNN']['val_acc'],color='g',label='CNN')
plt.plot(sg_imdb['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```



4 GloVe

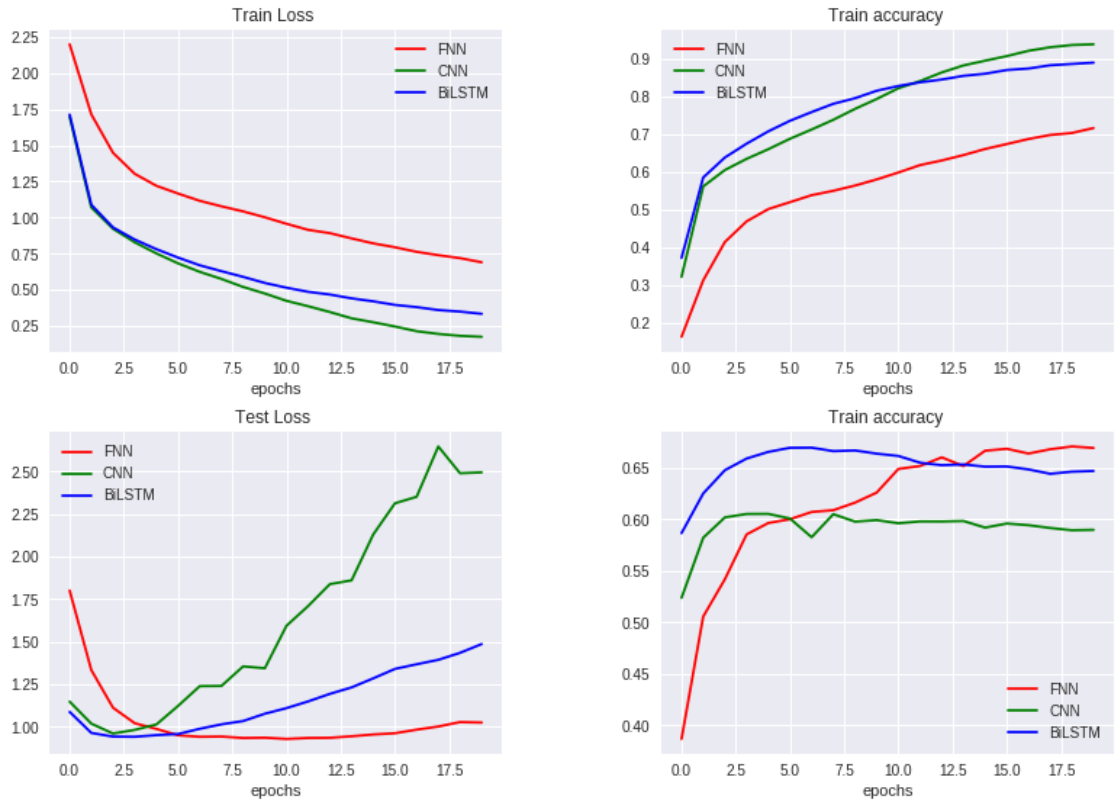
4.1 Customer Complaint

[0]:
```
g_cust = np.load(dep_path+'g_cust.npy').item()
g_cust.keys()
```

[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(g_cust['NN']['loss'],color='r',label='FNN')
plt.plot(g_cust['CNN']['loss'],color='g',label='CNN')
plt.plot(g_cust['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(g_cust['NN']['acc'],color='r',label='FNN')
plt.plot(g_cust['CNN']['acc'],color='g',label='CNN')
plt.plot(g_cust['BiLSTM']['acc'],color='b',label='BiLSTM')
```

```
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(g_cust['NN']['val_loss'],color='r',label='FNN')
plt.plot(g_cust['CNN']['val_loss'],color='g',label='CNN')
plt.plot(g_cust['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Train accuracy ')
plt.plot(g_cust['NN']['val_acc'],color='r',label='FNN')
plt.plot(g_cust['CNN']['val_acc'],color='g',label='CNN')
plt.plot(g_cust['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```

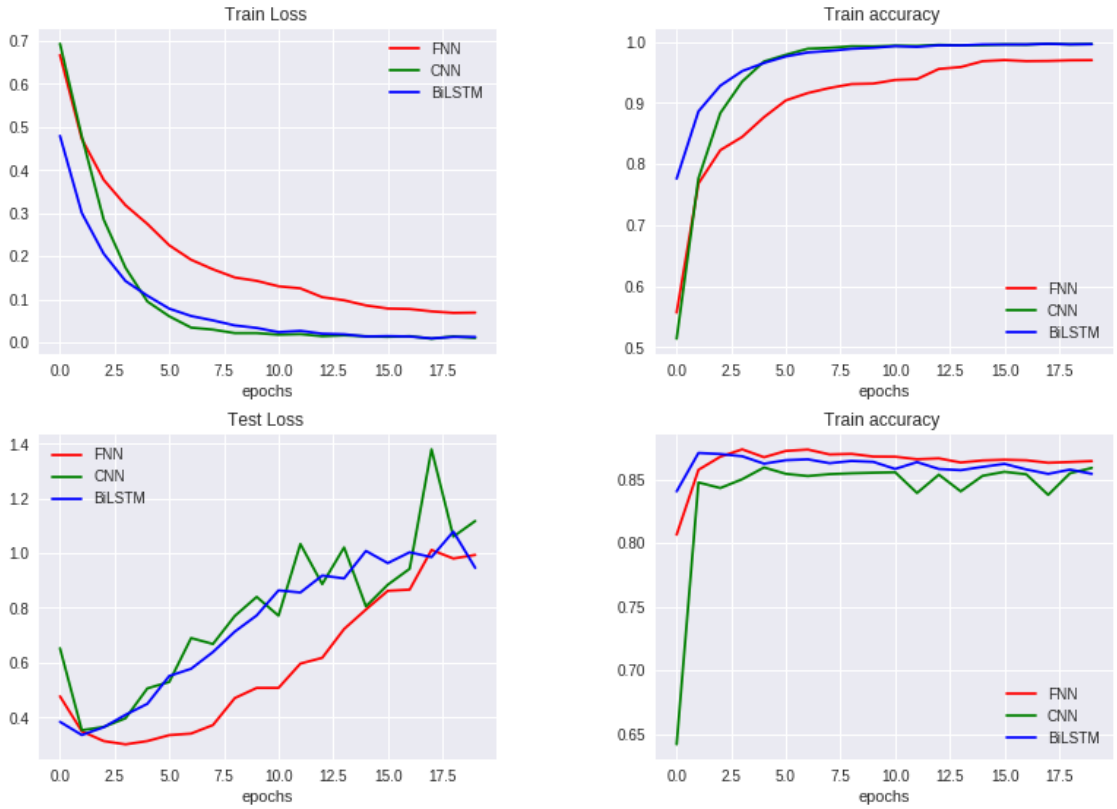


4.2 IMDB

[0]:
```
g_imdb = np.load(dep_path+'g_imdb.npy').item()
g_imdb.keys()
```

[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(g_imdb['NN']['loss'],color='r',label='FNN')
plt.plot(g_imdb['CNN']['loss'],color='g',label='CNN')
plt.plot(g_imdb['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(g_imdb['NN']['acc'],color='r',label='FNN')
plt.plot(g_imdb['CNN']['acc'],color='g',label='CNN')
plt.plot(g_imdb['BiLSTM']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(g_imdb['NN']['val_loss'],color='r',label='FNN')
plt.plot(g_imdb['CNN']['val_loss'],color='g',label='CNN')
plt.plot(g_imdb['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Train accuracy ')
plt.plot(g_imdb['NN']['val_acc'],color='r',label='FNN')
plt.plot(g_imdb['CNN']['val_acc'],color='g',label='CNN')
plt.plot(g_imdb['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```



5 FastText

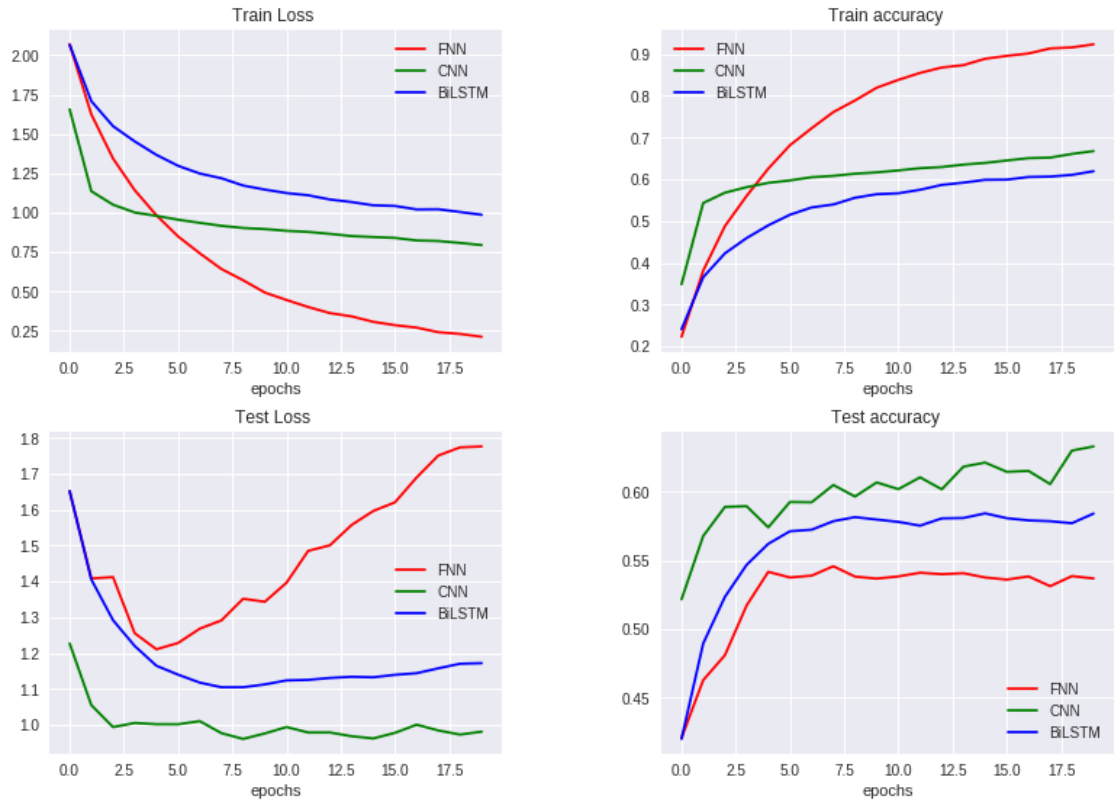
5.1 Customer complaint

[0]:
```
ft_cust = np.load(dep_path+'ft_cust.npy').item()
ft_cust.keys()
```

[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(ft_cust['NN']['loss'],color='r',label='FNN')
plt.plot(ft_cust['CNN']['loss'],color='g',label='CNN')
plt.plot(ft_cust['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(ft_cust['NN']['acc'],color='r',label='FNN')
plt.plot(ft_cust['CNN']['acc'],color='g',label='CNN')
plt.plot(ft_cust['BiLSTM']['acc'],color='b',label='BiLSTM')
```

```
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(ft_cust['NN']['val_loss'],color='r',label='FNN')
plt.plot(ft_cust['CNN']['val_loss'],color='g',label='CNN')
plt.plot(ft_cust['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Test accuracy ')
plt.plot(ft_cust['NN']['val_acc'],color='r',label='FNN')
plt.plot(ft_cust['CNN']['val_acc'],color='g',label='CNN')
plt.plot(ft_cust['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```

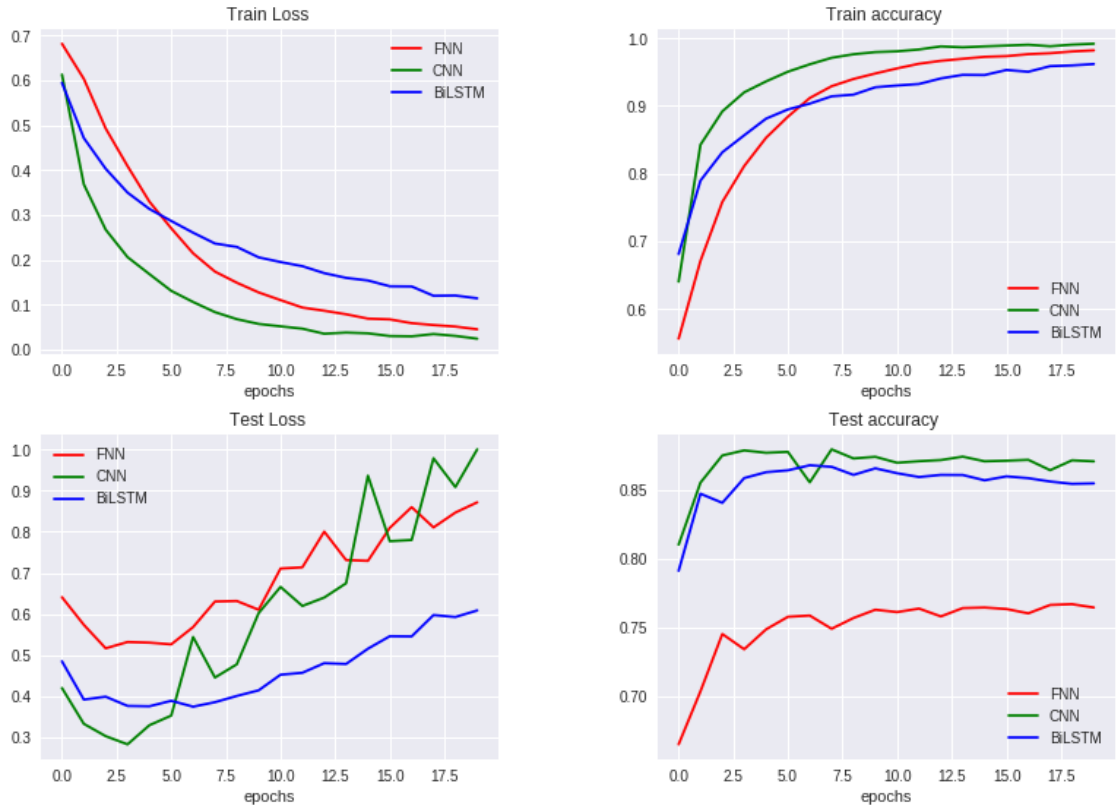


5.2 IMDB

[0]:
```
ft_imdb = np.load(dep_path+'ft_imdb.npy').item()
ft_imdb.keys()
```

[0]:
```
dict_keys(['NN', 'CNN', 'BiLSTM'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(ft_imdb['NN']['loss'],color='r',label='FNN')
plt.plot(ft_imdb['CNN']['loss'],color='g',label='CNN')
plt.plot(ft_imdb['BiLSTM']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(ft_imdb['NN']['acc'],color='r',label='FNN')
plt.plot(ft_imdb['CNN']['acc'],color='g',label='CNN')
plt.plot(ft_imdb['BiLSTM']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(ft_imdb['NN']['val_loss'],color='r',label='FNN')
plt.plot(ft_imdb['CNN']['val_loss'],color='g',label='CNN')
plt.plot(ft_imdb['BiLSTM']['val_loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Test accuracy ')
plt.plot(ft_imdb['NN']['val_acc'],color='r',label='FNN')
plt.plot(ft_imdb['CNN']['val_acc'],color='g',label='CNN')
plt.plot(ft_imdb['BiLSTM']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```



#CV

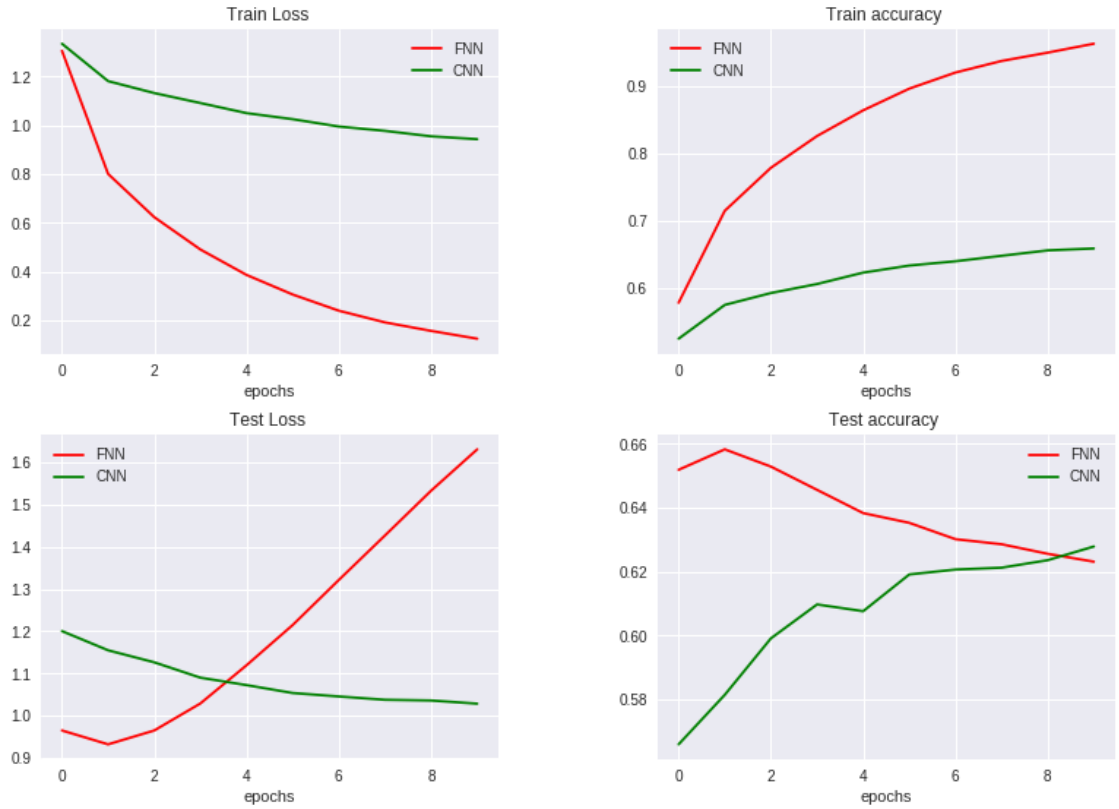
5.3 Customer Complaint

[0]:
```
cv_cust = np.load(dep_path+'cv_cust.npy').item()
cv_cust.keys()
```

[0]:
```
dict_keys(['NB', 'NN', 'CNN'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(cv_cust['NN']['loss'],color='r',label='FNN')
plt.plot(cv_cust['CNN']['loss'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(cv_cust['NN']['acc'],color='r',label='FNN')
plt.plot(cv_cust['CNN']['acc'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
```

```
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(cv_cust['NN']['val_loss'],color='r',label='FNN')
plt.plot(cv_cust['CNN']['val_loss'],color='g',label='CNN')
#plt.plot(cv_imdb['NB'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Test accuracy ')
plt.plot(cv_cust['NN']['val_acc'],color='r',label='FNN')
plt.plot(cv_cust['CNN']['val_acc'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```

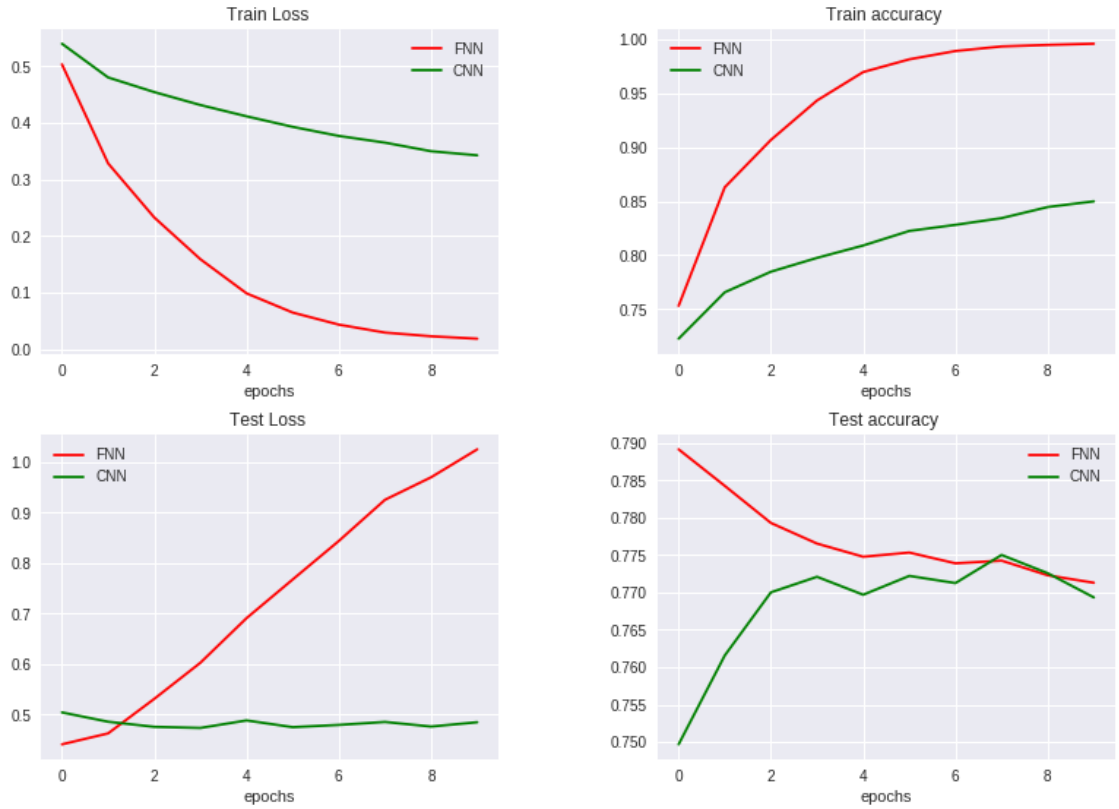


##IMDB

[0]:
```
cv_imdb = np.load(dep_path+'cv_imdb.npy').item()
cv_imdb.keys()
```

[0]:
```
dict_keys(['NB', 'NN', 'CNN'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(cv_imdb['NN']['loss'],color='r',label='FNN')
plt.plot(cv_imdb['CNN']['loss'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(cv_imdb['NN']['acc'],color='r',label='FNN')
plt.plot(cv_imdb['CNN']['acc'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(cv_imdb['NN']['val_loss'],color='r',label='FNN')
plt.plot(cv_imdb['CNN']['val_loss'],color='g',label='CNN')
#plt.plot(cv_imdb['NB'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Test accuracy ')
plt.plot(cv_imdb['NN']['val_acc'],color='r',label='FNN')
plt.plot(cv_imdb['CNN']['val_acc'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```



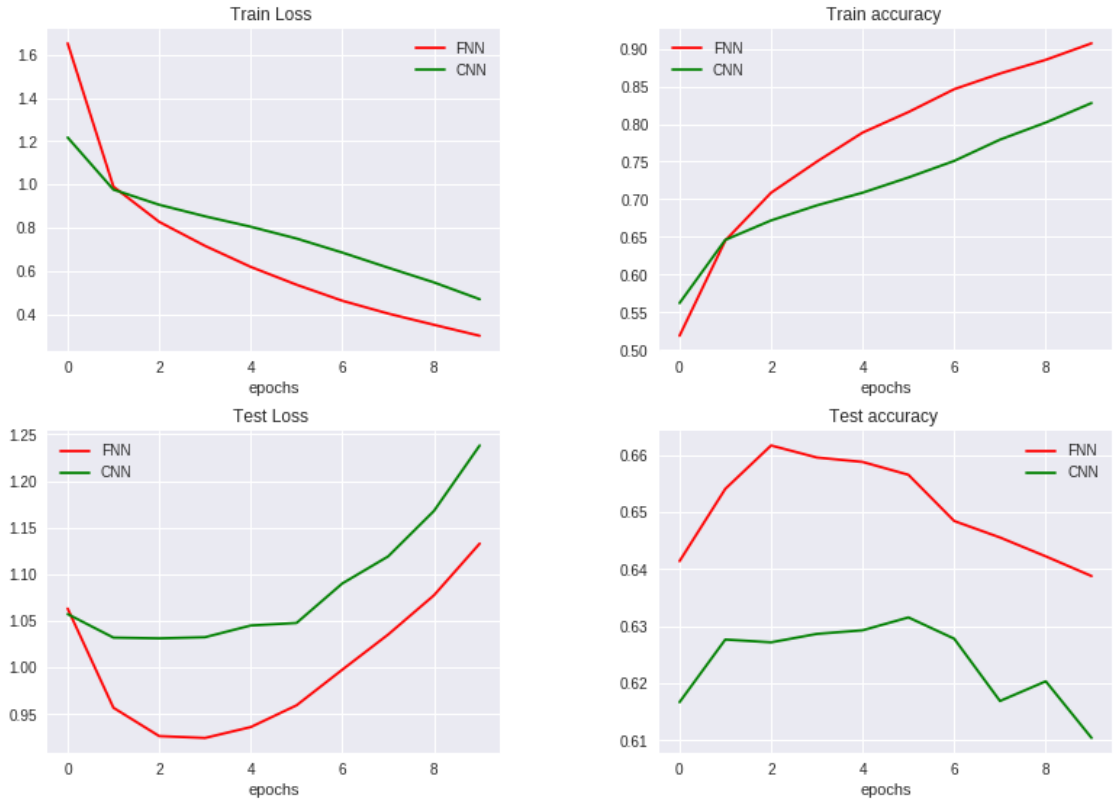
#TFIDF
##Customer Complaint

[0]:
```
tf_cust = np.load(dep_path+'tf_cust.npy').item()
tf_cust.keys()
```

[0]:
```
dict_keys(['NB', 'NN', 'CNN'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(tf_cust['NN']['loss'],color='r',label='FNN')
plt.plot(tf_cust['CNN']['loss'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(tf_cust['NN']['acc'],color='r',label='FNN')
plt.plot(tf_cust['CNN']['acc'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
```

```
plt.title('Test Loss ')
plt.plot(tf_cust['NN']['val_loss'],color='r',label='FNN')
plt.plot(tf_cust['CNN']['val_loss'],color='g',label='CNN')
#plt.plot(cv_imdb['NB'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Test accuracy ')
plt.plot(tf_cust['NN']['val_acc'],color='r',label='FNN')
plt.plot(tf_cust['CNN']['val_acc'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```

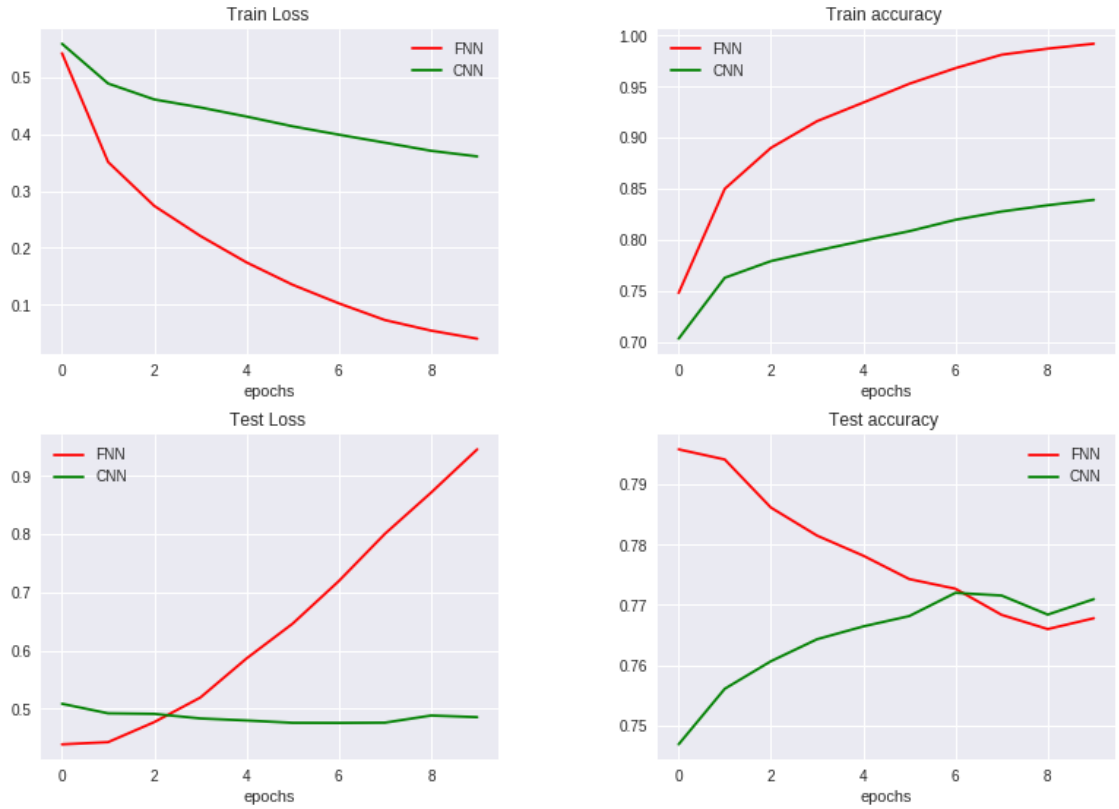


5.4 IMDB

[0]:
```
tf_imdb = np.load(dep_path+'tf_imdb.npy').item()
tf_imdb.keys()
```

[0]:
```
dict_keys(['NB', 'NN', 'CNN'])
```

[0]:
```
fig1 = plt.gcf()
plt.subplot(221)
plt.title('Train Loss ')
plt.plot(tf_imdb['NN']['loss'],color='r',label='FNN')
plt.plot(tf_imdb['CNN']['loss'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['loss'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(222)
plt.title('Train accuracy ')
plt.plot(tf_imdb['NN']['acc'],color='r',label='FNN')
plt.plot(tf_imdb['CNN']['acc'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(223)
plt.title('Test Loss ')
plt.plot(tf_imdb['NN']['val_loss'],color='r',label='FNN')
plt.plot(tf_imdb['CNN']['val_loss'],color='g',label='CNN')
#plt.plot(cv_imdb['NB'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplot(224)
plt.title('Test accuracy ')
plt.plot(tf_imdb['NN']['val_acc'],color='r',label='FNN')
plt.plot(tf_imdb['CNN']['val_acc'],color='g',label='CNN')
#plt.plot(cv_imdb['NB']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.35)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*1.5, DefaultSize[1]*1.5) )
```



[0]:
```
import pandas as pd
r = ['Method','NB','FNN','CNN','RNN']
tf_i = ['TF-IDF',tf_imdb['NB'],np.array(tf_imdb['NN']['val_acc']).argmax(),np.
 ↪array(tf_imdb['CNN']['val_acc']).argmax(),"NA"]
tf_c = ['TF-IDF',tf_cust['NB'],np.array(tf_cust['NN']['val_acc']).argmax(),np.
 ↪array(tf_cust['CNN']['val_acc']).argmax(),"NA"]
cv_i = ['Count Vector',cv_imdb['NB'],np.array(cv_imdb['NN']['val_acc']).
 ↪argmax(),np.array(cv_imdb['CNN']['val_acc']).argmax(),"NA"]
cv_c = ['Count Vector',cv_cust['NB'],np.array(cv_cust['NN']['val_acc']).
 ↪argmax(),np.array(cv_cust['CNN']['val_acc']).argmax(),"NA"]
```

[0]:
```
pre_i = ['Pre-Trained',"NA",np.array(pre_imdb['NN']['val_acc']).argmax(),np.
 ↪array(pre_imdb['CNN']['val_acc']).argmax(),np.
 ↪array(pre_imdb['BiLSTM']['val_acc']).argmax()]
pre_c = ['Pre-Trained',"NA",np.array(pre_cust['NN']['val_acc']).argmax(),np.
 ↪array(pre_cust['CNN']['val_acc']).argmax(),np.
 ↪array(pre_cust['BiLSTM']['val_acc']).argmax()]
c_i = ['CBOW',"NA",np.array(cbow_imdb['NN']['val_acc']).argmax(),np.
 ↪array(cbow_imdb['CNN']['val_acc']).argmax(),np.
 ↪array(cbow_imdb['BiLSTM']['val_acc']).argmax()]
```

```
c_c = ['CBOW',"NA",np.array(cbow_cust['NN']['val_acc']).argmax(),np.
 ↪array(cbow_cust['CNN']['val_acc']).argmax(),np.
 ↪array(cbow_cust['BiLSTM']['val_acc']).argmax()]
s_i = ['Skip-Gram',"NA",np.array(sg_imdb['NN']['val_acc']).argmax(),np.
 ↪array(sg_imdb['CNN']['val_acc']).argmax(),np.
 ↪array(sg_imdb['BiLSTM']['val_acc']).argmax()]
s_c = ['Skip-Gram',"NA",np.array(sg_cust['NN']['val_acc']).argmax(),np.
 ↪array(sg_cust['CNN']['val_acc']).argmax(),np.
 ↪array(sg_cust['BiLSTM']['val_acc']).argmax()]
g_i = ['GloVe',"NA",np.array(g_imdb['NN']['val_acc']).argmax(),np.
 ↪array(g_imdb['CNN']['val_acc']).argmax(),np.
 ↪array(g_imdb['BiLSTM']['val_acc']).argmax()]
g_c = ['GloVe',"NA",np.array(g_cust['NN']['val_acc']).argmax(),np.
 ↪array(g_cust['CNN']['val_acc']).argmax(),np.
 ↪array(g_cust['BiLSTM']['val_acc']).argmax()]
f_i = ['FastText',"NA",np.array(ft_imdb['NN']['val_acc']).argmax(),np.
 ↪array(ft_imdb['CNN']['val_acc']).argmax(),np.
 ↪array(ft_imdb['BiLSTM']['val_acc']).argmax()]
f_c = ['FastText',"NA",np.array(ft_cust['NN']['val_acc']).argmax(),np.
 ↪array(ft_cust['CNN']['val_acc']).argmax(),np.
 ↪array(ft_cust['BiLSTM']['val_acc']).argmax()]
```

[0]:
```
imdb_p = pd.DataFrame([cv_i,tf_i,pre_i,c_i,s_i,g_i,f_i],columns = r)
cust_p = pd.DataFrame([cv_c,tf_c,pre_c,c_c,s_c,g_c,f_c],columns = r)

print("IMDB dataset Results")
imdb_p
```

```
IMDB dataset Results
```

[0]:

	Method	NB	FNN	CNN	RNN
0	Count Vector	78.624	0	7	NA
1	TF-IDF	78.78	0	6	NA
2	Pre-Trained	NA	11	1	7
3	CBOW	NA	19	2	5
4	Skip-Gram	NA	18	0	3
5	GloVe	NA	3	4	1
6	FastText	NA	18	7	6

[0]:
```
print("Customer complaint Dataset Results")
cust_p
```

```
Customer complaint Dataset Results
```

[0]:

	Method	NB	FNN	CNN	RNN
0	Count Vector	64.1039	1	9	NA
1	TF-IDF	64.4194	2	5	NA

```
2   Pre-Trained       NA   17   12   4
3          CBOW       NA    9   15  12
4     Skip-Gram       NA   19   16   5
5         GloVe       NA   18    4   6
6      FastText       NA    7   19  14
```

5.5 accuracy values

[0]:
```
import pandas as pd
r = ['Method','NB','FNN','CNN','RNN']
tf_i = ['TF-IDF',tf_imdb['NB'],np.array(tf_imdb['NN']['val_acc']).max()*100,np.
 ↪array(tf_imdb['CNN']['val_acc']).max()*100,"NA"]
tf_c = ['TF-IDF',tf_cust['NB'],np.array(tf_cust['NN']['val_acc']).max()*100,np.
 ↪array(tf_cust['CNN']['val_acc']).max()*100,"NA"]
cv_i = ['Count Vector',cv_imdb['NB'],np.array(cv_imdb['NN']['val_acc']).
 ↪max()*100,np.array(cv_imdb['CNN']['val_acc']).max()*100,"NA"]
cv_c = ['Count Vector',cv_cust['NB'],np.array(cv_cust['NN']['val_acc']).
 ↪max()*100,np.array(cv_cust['CNN']['val_acc']).max()*100,"NA"]
```

[0]:
```
pre_i = ['Pre-Trained',"NA",np.array(pre_imdb['NN']['val_acc']).max()*100,np.
 ↪array(pre_imdb['CNN']['val_acc']).max()*100,np.
 ↪array(pre_imdb['BiLSTM']['val_acc']).max()*100]
pre_c = ['Pre-Trained',"NA",np.array(pre_cust['NN']['val_acc']).max()*100,np.
 ↪array(pre_cust['CNN']['val_acc']).max()*100,np.
 ↪array(pre_cust['BiLSTM']['val_acc']).max()*100]
c_i = ['CBOW',"NA",np.array(cbow_imdb['NN']['val_acc']).max()*100,np.
 ↪array(cbow_imdb['CNN']['val_acc']).max()*100,np.
 ↪array(cbow_imdb['BiLSTM']['val_acc']).max()*100]
c_c = ['CBOW',"NA",np.array(cbow_cust['NN']['val_acc']).max()*100,np.
 ↪array(cbow_cust['CNN']['val_acc']).max()*100,np.
 ↪array(cbow_cust['BiLSTM']['val_acc']).max()*100]
s_i = ['Skip-Gram',"NA",np.array(sg_imdb['NN']['val_acc']).max()*100,np.
 ↪array(sg_imdb['CNN']['val_acc']).max()*100,np.
 ↪array(sg_imdb['BiLSTM']['val_acc']).max()*100]
s_c = ['Skip-Gram',"NA",np.array(sg_cust['NN']['val_acc']).max()*100,np.
 ↪array(sg_cust['CNN']['val_acc']).max()*100,np.
 ↪array(sg_cust['BiLSTM']['val_acc']).max()*100]
g_i = ['GloVe',"NA",np.array(g_imdb['NN']['val_acc']).max()*100,np.
 ↪array(g_imdb['CNN']['val_acc']).max()*100,np.
 ↪array(g_imdb['BiLSTM']['val_acc']).max()*100]
g_c = ['GloVe',"NA",np.array(g_cust['NN']['val_acc']).max()*100,np.
 ↪array(g_cust['CNN']['val_acc']).max()*100,np.
 ↪array(g_cust['BiLSTM']['val_acc']).max()*100]
f_i = ['FastText',"NA",np.array(ft_imdb['NN']['val_acc']).max()*100,np.
 ↪array(ft_imdb['CNN']['val_acc']).max()*100,np.
 ↪array(ft_imdb['BiLSTM']['val_acc']).max()*100]
```

```
f_c = ['FastText',"NA",np.array(ft_cust['NN']['val_acc']).max()*100,np.
 ↪array(ft_cust['CNN']['val_acc']).max()*100,np.
 ↪array(ft_cust['BiLSTM']['val_acc']).max()*100]
```

[0]:
```
imdb = pd.DataFrame([cv_i,tf_i,pre_i,c_i,s_i,g_i,f_i],columns = r)
cust = pd.DataFrame([cv_c,tf_c,pre_c,c_c,s_c,g_c,f_c],columns = r)
```

[0]:
```
print("IMDB dataset Results")
imdb
```

```
IMDB dataset Results
```

[0]:

	Method	NB	FNN	CNN	RNN
0	Count Vector	78.624	78.912	77.500	NA
1	TF-IDF	78.78	79.576	77.200	NA
2	Pre-Trained	NA	87.200	87.412	83.656
3	CBOW	NA	70.132	88.268	88.36
4	Skip-Gram	NA	86.664	87.836	86.172
5	GloVe	NA	87.352	85.928	87.068
6	FastText	NA	76.684	87.960	86.8

[0]:
```
print("Customer complaint Dataset Results")
cust
```

```
Customer complaint Dataset Results
```

[0]:

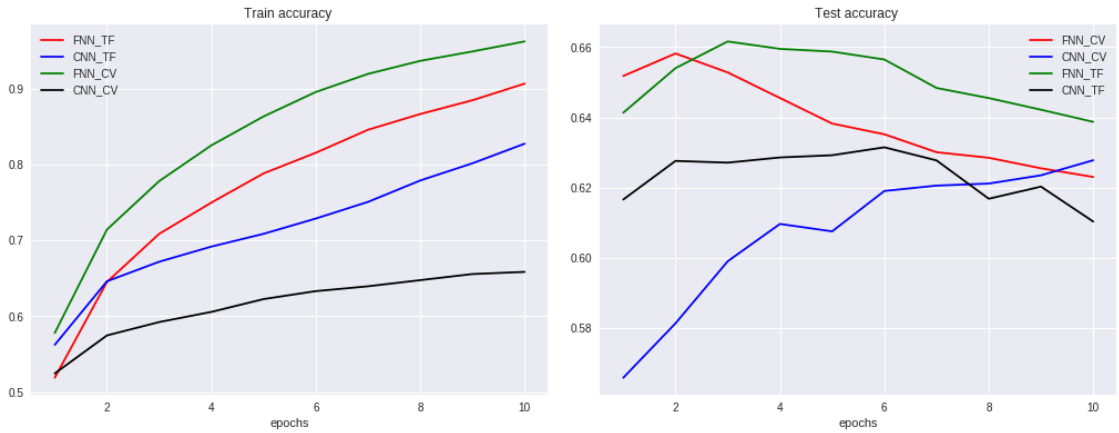
	Method	NB	FNN	CNN	RNN
0	Count Vector	64.1039	65.822191	62.784424	NA
1	TF-IDF	64.4194	66.164673	63.152709	NA
2	Pre-Trained	NA	67.037298	62.786770	66.5869
3	CBOW	NA	36.699507	68.605510	65.1935
4	Skip-Gram	NA	59.486277	61.745250	66.2843
5	GloVe	NA	67.095607	60.540867	66.98
6	FastText	NA	54.567206	63.285126	58.4166

[0]:
```
from google.colab import files
imdb.to_csv('IMDB.csv')
#files.download('IMDB.csv')
cust.to_csv('CUST.csv')
#files.download('CUST.csv')
```

6 GRAPHS OUTPUTS

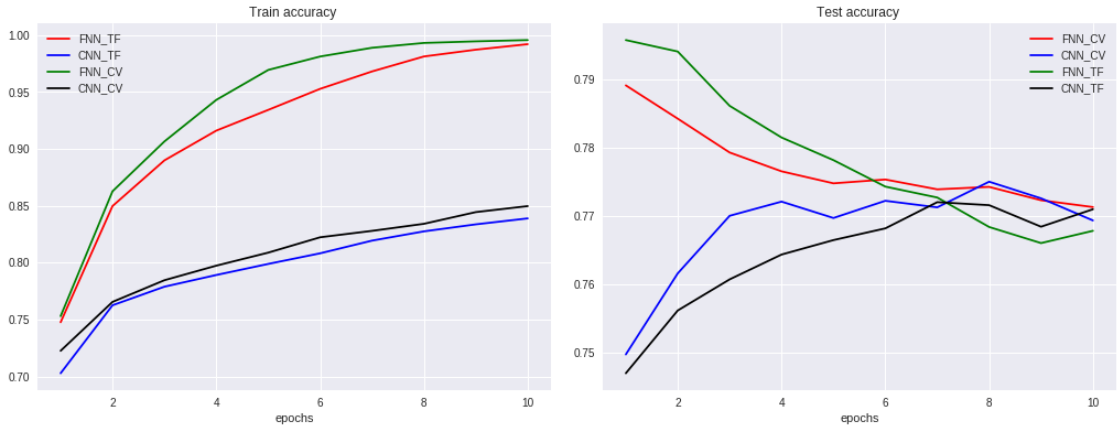
6.1 CUST Frequency Based

[0]:
```
fig1 = plt.gcf()
plt.subplot(121)
y = list(range(1,len(tf_cust['CNN']['acc'])+1))
plt.title('Train accuracy ')
plt.plot(y,tf_cust['NN']['acc'],color='r',label='FNN_TF')
plt.plot(y,tf_cust['CNN']['acc'],color='b',label='CNN_TF')
plt.plot(y,cv_cust['NN']['acc'],color='g',label='FNN_CV')
plt.plot(y,cv_cust['CNN']['acc'],color='black',label='CNN_CV')
#plt.plot(cv_imdb['NB']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()

plt.subplot(122)
plt.title('Test accuracy ')
plt.plot(y,cv_cust['NN']['val_acc'],color='r',label='FNN_CV')
plt.plot(y,cv_cust['CNN']['val_acc'],color='b',label='CNN_CV')
plt.plot(y,tf_cust['NN']['val_acc'],color='g',label='FNN_TF')
plt.plot(y,tf_cust['CNN']['val_acc'],color='black',label='CNN_TF')
#plt.plot(cv_imdb['NB']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.10)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*2, DefaultSize[1]*1) )
```



6.2 IMDB Frequecy Based

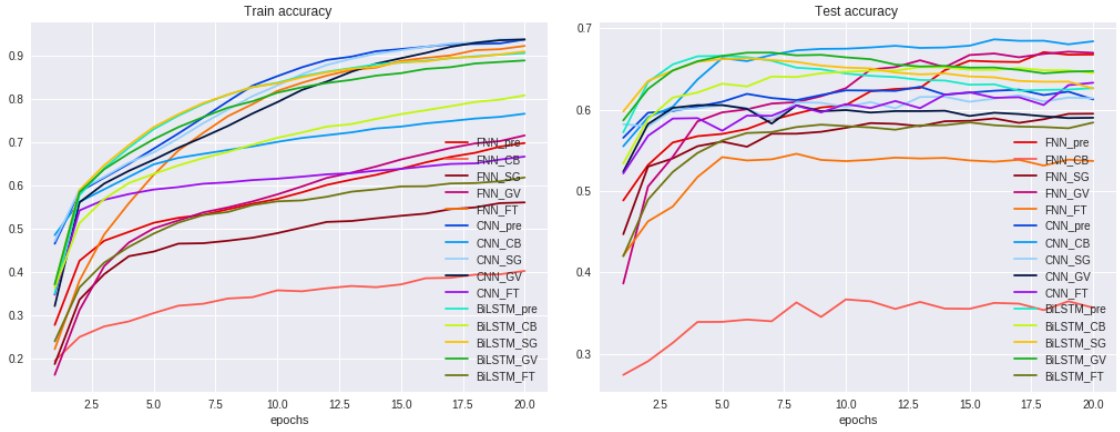
[0]:
```
fig1 = plt.gcf()
plt.subplot(121)
y = list(range(1,len(tf_imdb['CNN']['acc'])+1))
plt.title('Train accuracy ')
plt.plot(y,tf_imdb['NN']['acc'],color='r',label='FNN_TF')
plt.plot(y,tf_imdb['CNN']['acc'],color='b',label='CNN_TF')
plt.plot(y,cv_imdb['NN']['acc'],color='g',label='FNN_CV')
plt.plot(y,cv_imdb['CNN']['acc'],color='black',label='CNN_CV')
#plt.plot(cv_imdb['NB']['acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()

plt.subplot(122)
plt.title('Test accuracy ')
plt.plot(y,cv_imdb['NN']['val_acc'],color='r',label='FNN_CV')
plt.plot(y,cv_imdb['CNN']['val_acc'],color='b',label='CNN_CV')
plt.plot(y,tf_imdb['NN']['val_acc'],color='g',label='FNN_TF')
plt.plot(y,tf_imdb['CNN']['val_acc'],color='black',label='CNN_TF')
#plt.plot(cv_imdb['NB']['val_acc'],color='b',label='BiLSTM')
plt.xlabel('epochs')
plt.legend()
plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.10)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*2, DefaultSize[1]*1) )
```



6.3 CUST PREDICTIVE based

[0]:
```
fig1 = plt.gcf()
y = list(range(1,len(pre_cust['CNN']['acc'])+1))
r = ['xkcd:red','xkcd:coral','xkcd:crimson','xkcd:magenta','xkcd:orange']
b = ['xkcd:blue','xkcd:azure','xkcd:light blue','xkcd:navy','xkcd:violet']
g = ['xkcd:aqua','xkcd:chartreuse','xkcd:goldenrod','xkcd:green','xkcd:olive']
color=iter((r+b+g))
plt.subplot(121)
plt.title('Train accuracy ')
plt.plot(y,pre_cust['NN']['acc'],color=next(color),label='FNN_pre')
plt.plot(y,cbow_cust['NN']['acc'],color=next(color),label='FNN_CB')
plt.plot(y,sg_cust['NN']['acc'],color=next(color),label='FNN_SG')
plt.plot(y,g_cust['NN']['acc'],color=next(color),label='FNN_GV')
plt.plot(y,ft_cust['NN']['acc'],color=next(color),label='FNN_FT')

plt.plot(y,pre_cust['CNN']['acc'],color=next(color),label='CNN_pre')
plt.plot(y,cbow_cust['CNN']['acc'],color=next(color),label='CNN_CB')
plt.plot(y,sg_cust['CNN']['acc'],color=next(color),label='CNN_SG')
plt.plot(y,g_cust['CNN']['acc'],color=next(color),label='CNN_GV')
plt.plot(y,ft_cust['CNN']['acc'],color=next(color),label='CNN_FT')

plt.plot(y,pre_cust['BiLSTM']['acc'],color=next(color),label='BiLSTM_pre')
plt.plot(y,cbow_cust['BiLSTM']['acc'],color=next(color),label='BiLSTM_CB')
plt.plot(y,sg_cust['BiLSTM']['acc'],color=next(color),label='BiLSTM_SG')
plt.plot(y,g_cust['BiLSTM']['acc'],color=next(color),label='BiLSTM_GV')
plt.plot(y,ft_cust['BiLSTM']['acc'],color=next(color),label='BiLSTM_FT')

plt.xlabel('epochs')
plt.legend()
color=iter((r+b+g))
plt.subplot(122)
plt.title('Test accuracy ')
plt.plot(y,pre_cust['NN']['val_acc'],color=next(color),label='FNN_pre')
plt.plot(y,cbow_cust['NN']['val_acc'],color=next(color),label='FNN_CB')
plt.plot(y,sg_cust['NN']['val_acc'],color=next(color),label='FNN_SG')
plt.plot(y,g_cust['NN']['val_acc'],color=next(color),label='FNN_GV')
plt.plot(y,ft_cust['NN']['val_acc'],color=next(color),label='FNN_FT')

plt.plot(y,pre_cust['CNN']['val_acc'],color=next(color),label='CNN_pre')
plt.plot(y,cbow_cust['CNN']['val_acc'],color=next(color),label='CNN_CB')
plt.plot(y,sg_cust['CNN']['val_acc'],color=next(color),label='CNN_SG')
plt.plot(y,g_cust['CNN']['val_acc'],color=next(color),label='CNN_GV')
plt.plot(y,ft_cust['CNN']['val_acc'],color=next(color),label='CNN_FT')

plt.plot(y,pre_cust['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_pre')
plt.plot(y,cbow_cust['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_CB')
```

```
plt.plot(y,sg_cust['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_SG')
plt.plot(y,g_cust['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_GV')
plt.plot(y,ft_cust['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_FT')

plt.xlabel('epochs')
plt.legend()

plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.10)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*2, DefaultSize[1]*1) )
```

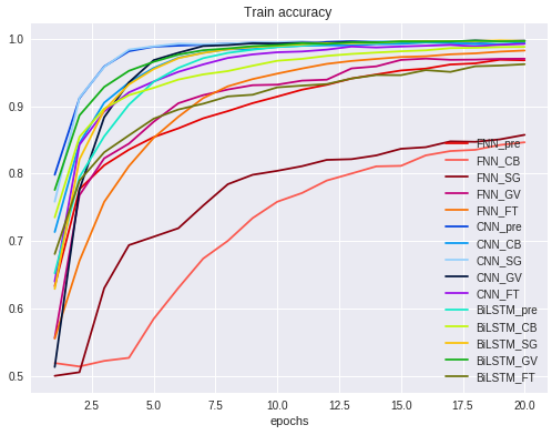


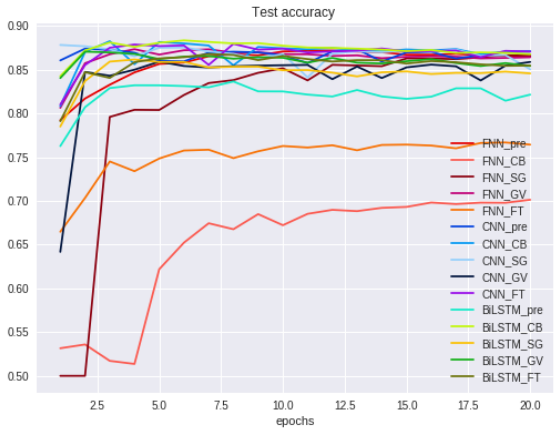
[0]:

6.4 IMDB PREDICTIVE based

[0]:
```
fig1 = plt.gcf()
y = list(range(1,len(pre_imdb['CNN']['acc'])+1))
r = ['xkcd:red','xkcd:coral','xkcd:crimson','xkcd:magenta','xkcd:orange']
b = ['xkcd:blue','xkcd:azure','xkcd:light blue','xkcd:navy','xkcd:violet']
g = ['xkcd:aqua','xkcd:chartreuse','xkcd:goldenrod','xkcd:green','xkcd:olive']
color=iter((r+b+g))
plt.subplot(121)
plt.title('Train accuracy ')
plt.plot(y,pre_imdb['NN']['acc'],color=next(color),label='FNN_pre')
plt.plot(y,cbow_imdb['NN']['acc'],color=next(color),label='FNN_CB')
plt.plot(y,sg_imdb['NN']['acc'],color=next(color),label='FNN_SG')
plt.plot(y,g_imdb['NN']['acc'],color=next(color),label='FNN_GV')
plt.plot(y,ft_imdb['NN']['acc'],color=next(color),label='FNN_FT')

plt.plot(y,pre_imdb['CNN']['acc'],color=next(color),label='CNN_pre')
plt.plot(y,cbow_imdb['CNN']['acc'],color=next(color),label='CNN_CB')
```

```
plt.plot(y,sg_imdb['CNN']['acc'],color=next(color),label='CNN_SG')
plt.plot(y,g_imdb['CNN']['acc'],color=next(color),label='CNN_GV')
plt.plot(y,ft_imdb['CNN']['acc'],color=next(color),label='CNN_FT')

plt.plot(y,pre_imdb['BiLSTM']['acc'],color=next(color),label='BiLSTM_pre')
plt.plot(y,cbow_imdb['BiLSTM']['acc'],color=next(color),label='BiLSTM_CB')
plt.plot(y,sg_imdb['BiLSTM']['acc'],color=next(color),label='BiLSTM_SG')
plt.plot(y,g_imdb['BiLSTM']['acc'],color=next(color),label='BiLSTM_GV')
plt.plot(y,ft_imdb['BiLSTM']['acc'],color=next(color),label='BiLSTM_FT')

plt.xlabel('epochs')
plt.legend()

color=iter((r+b+g))
plt.subplot(122)
plt.title('Test accuracy ')
plt.plot(y,pre_imdb['NN']['val_acc'],color=next(color),label='FNN_pre')
plt.plot(y,cbow_imdb['NN']['val_acc'],color=next(color),label='FNN_CB')
plt.plot(y,sg_imdb['NN']['val_acc'],color=next(color),label='FNN_SG')
plt.plot(y,g_imdb['NN']['val_acc'],color=next(color),label='FNN_GV')
plt.plot(y,ft_imdb['NN']['val_acc'],color=next(color),label='FNN_FT')

plt.plot(y,pre_imdb['CNN']['val_acc'],color=next(color),label='CNN_pre')
plt.plot(y,cbow_imdb['CNN']['val_acc'],color=next(color),label='CNN_CB')
plt.plot(y,sg_imdb['CNN']['val_acc'],color=next(color),label='CNN_SG')
plt.plot(y,g_imdb['CNN']['val_acc'],color=next(color),label='CNN_GV')
plt.plot(y,ft_imdb['CNN']['val_acc'],color=next(color),label='CNN_FT')

plt.plot(y,pre_imdb['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_pre')
plt.plot(y,cbow_imdb['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_CB')
plt.plot(y,sg_imdb['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_SG')
plt.plot(y,g_imdb['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_GV')
plt.plot(y,ft_imdb['BiLSTM']['val_acc'],color=next(color),label='BiLSTM_FT')

plt.xlabel('epochs')
plt.legend()

plt.subplots_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95, hspace=0.25,
                    wspace=0.10)
DefaultSize = fig1.get_size_inches()
fig1.set_size_inches( (DefaultSize[0]*2, DefaultSize[1]*1) )
```


Train accuracy
1.0
0.9
0.8
0.7
0.6
0.5
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
epochs
FNN_pre
FNN_CB
FNN_SG
FNN_GV
FNN_FT
CNN_pre
CNN_CB
CNN_SG
CNN_GV
CNN_FT
BiLSTM_pre
BiLSTM_CB
BiLSTM_SG
BiLSTM_GV
BiLSTM_FT


Test accuracy
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
epochs
FNN_pre
FNN_CB
FNN_SG
FNN_GV
FNN_FT
CNN_pre
CNN_CB
CNN_SG
CNN_GV
CNN_FT
BiLSTM_pre
BiLSTM_CB
BiLSTM_SG
BiLSTM_GV
BiLSTM_FT