In [15]:

```
import tensorflow as tf
```

In [16]:

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense,Dropout,Activation,Flatten,Conv1D
from tensorflow.keras.optimizers import Adam
```

In [17]:

```
from sklearn import datasets
cancerdata=datasets.load_breast_cancer()
```

In [18]:

```
import numpy as np
import pandas as pd
x=pd.DataFrame(data=cancerdata.data,columns=cancerdata.feature_names)
x.head()
```

Out[18]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	me symme
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.24
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.18
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.20
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.25
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.18

5 rows × 30 columns

In [19]:
```
y=cancerdata.target
y
```

Out[19]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0,
       0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
       1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1,
       1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1])
```

In [20]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(512, 30)
(57, 30)
(512,)
(57,)
```

In [21]:
```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
x_train=scaler.fit_transform(x_train)
x_test=scaler.fit_transform(x_test)
```

In [22]:
```
x_train=x_train.reshape(512,30,1)
x_test=x_test.reshape(57,30,1)
```

In [27]:
```
model=Sequential()
model.add(Conv1D(16,2,activation='relu',input_shape=(30,1)))
model.add(Dropout(0.2))
model.add(Conv1D(32,2,activation='relu'))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(32,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1,activation='sigmoid'))
```

In [28]:
```
model.compile(optimizer=Adam(learning_rate=0.0001),loss='binary_crossentrop
```

In [29]:
```
tf_callbacks=tf.keras.callbacks.TensorBoard(log_dir='logs/fit',histogram_fr
```

In [30]:
```
history=model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=10
```

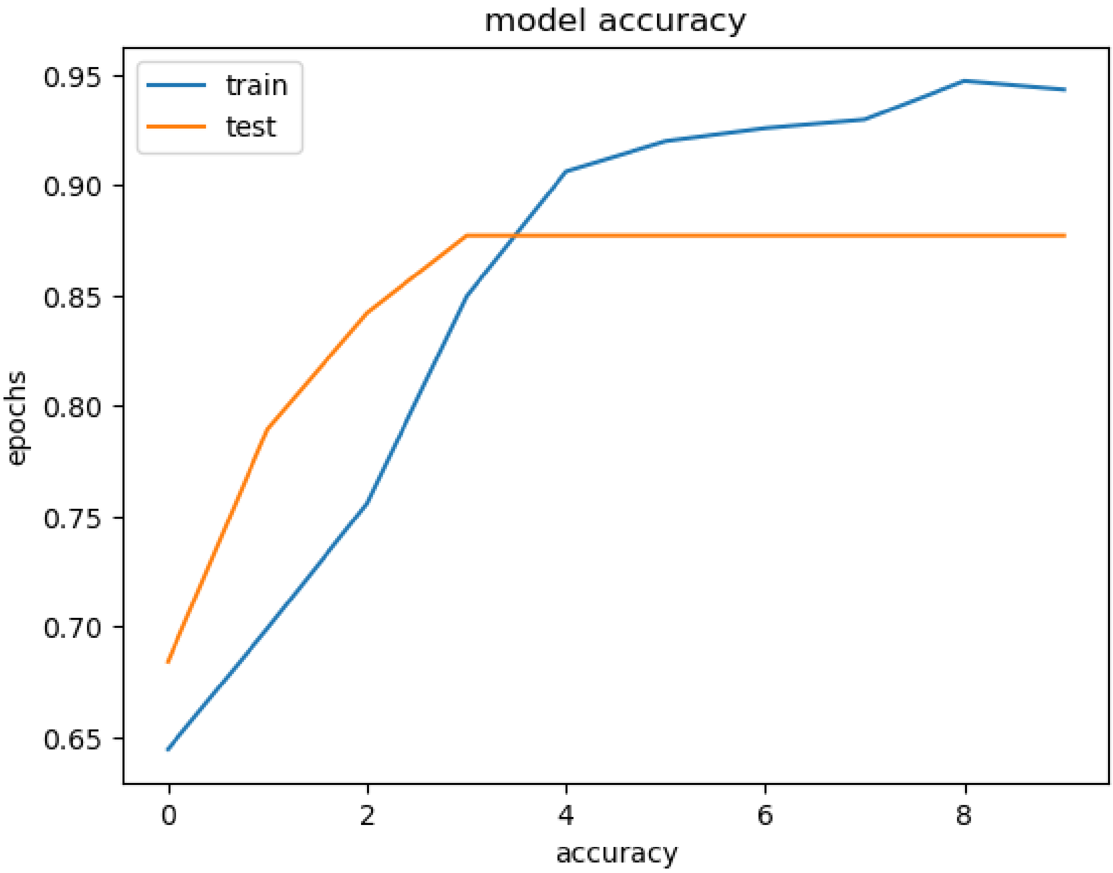
```
Epoch 1/10
16/16 [==============================] - 3s 52ms/step - loss: 0.6269 - acc
uracy: 0.6445 - val_loss: 0.5945 - val_accuracy: 0.6842
Epoch 2/10
16/16 [==============================] - 0s 23ms/step - loss: 0.5420 - acc
uracy: 0.6992 - val_loss: 0.5240 - val_accuracy: 0.7895
Epoch 3/10
16/16 [==============================] - 0s 23ms/step - loss: 0.4802 - acc
uracy: 0.7559 - val_loss: 0.4687 - val_accuracy: 0.8421
Epoch 4/10
16/16 [==============================] - 0s 23ms/step - loss: 0.4238 - acc
uracy: 0.8496 - val_loss: 0.4241 - val_accuracy: 0.8772
Epoch 5/10
16/16 [==============================] - 0s 23ms/step - loss: 0.3702 - acc
uracy: 0.9062 - val_loss: 0.3876 - val_accuracy: 0.8772
Epoch 6/10
16/16 [==============================] - 0s 23ms/step - loss: 0.3359 - acc
uracy: 0.9199 - val_loss: 0.3577 - val_accuracy: 0.8772
Epoch 7/10
16/16 [==============================] - 0s 23ms/step - loss: 0.2958 - acc
uracy: 0.9258 - val_loss: 0.3336 - val_accuracy: 0.8772
Epoch 8/10
16/16 [==============================] - 0s 23ms/step - loss: 0.2664 - acc
uracy: 0.9297 - val_loss: 0.3147 - val_accuracy: 0.8772
Epoch 9/10
16/16 [==============================] - 0s 23ms/step - loss: 0.2390 - acc
uracy: 0.9473 - val_loss: 0.2982 - val_accuracy: 0.8772
Epoch 10/10
16/16 [==============================] - 0s 25ms/step - loss: 0.2201 - acc
uracy: 0.9434 - val_loss: 0.2834 - val_accuracy: 0.8772
```

In [ ]:
```
load_ext tensorboard
```

In [17]:
```
tensorboard --logdir logs/fit
```

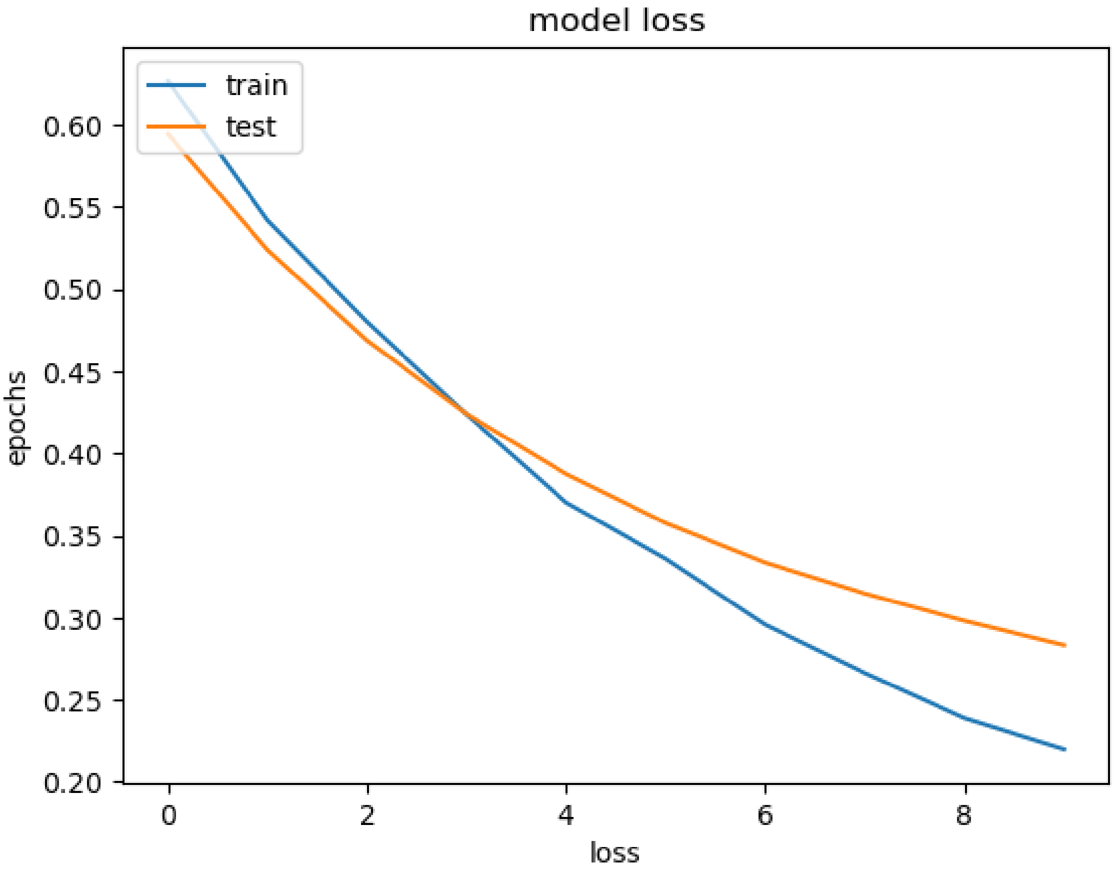
```
Reusing TensorBoard on port 6006 (pid 13432), started 22 days, 22:03:10 ag
o. (Use '!kill 13432' to kill it.)
```

In [31]:
```python
import matplotlib.pyplot as plt
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.xlabel('accuracy')
plt.ylabel('epochs')
plt.legend(['train','test'],loc='upper left')
plt.show()
```



In [32]:

```
import matplotlib.pyplot as plt
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.xlabel('loss')
plt.ylabel('epochs')
plt.legend(['train','test'],loc='upper left')
plt.show()
```



In [ ]: