

## practical-no-2-a

April 10, 2024

```
[1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

```
[2]: from keras.datasets import imdb
(X_train, y_train), (X_test, y_test) = imdb.load_data(num_words=10000)
```

WARNING:tensorflow:From C:\Users\harsh\Documents\anaconda\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse\_softmax\_cross\_entropy is deprecated. Please use tf.compat.v1.losses.sparse\_softmax\_cross\_entropy instead.

```
[3]: data = np.concatenate((X_train, X_test), axis=0)
```

```
[4]: label = np.concatenate((y_train, y_test), axis=0)
```

```
[5]: X_train.shape
```

```
[5]: (25000,)
```

```
[6]: X_test.shape
```

```
[6]: (25000,)
```

```
[7]: y_test.shape
```

```
[7]: (25000,)
```

```
[8]: y_train.shape
```

```
[8]: (25000,)
```

```
[9]: print("Review is ",X_train[0]) # series of no converted word to vocabulary
      ↪ associated with index
      print("Review is ",y_train[0])
```

Review is [1, 14, 22, 16, 43, 530, 973, 1622, 1385, 65, 458, 4468, 66, 3941, 4, 173, 36, 256, 5, 25, 100, 43, 838, 112, 50, 670, 2, 9, 35, 480, 284, 5, 150, 4, 172, 112, 167, 2, 336, 385, 39, 4, 172, 4536, 1111, 17, 546, 38, 13, 447, 4, 192, 50, 16, 6, 147, 2025, 19, 14, 22, 4, 1920, 4613, 469, 4, 22, 71, 87, 12, 16, 43, 530, 38, 76, 15, 13, 1247, 4, 22, 17, 515, 17, 12, 16, 626, 18, 2, 5, 62, 386, 12, 8, 316, 8, 106, 5, 4, 2223, 5244, 16, 480, 66, 3785, 33, 4, 130, 12, 16, 38, 619, 5, 25, 124, 51, 36, 135, 48, 25, 1415, 33, 6, 22, 12, 215, 28, 77, 52, 5, 14, 407, 16, 82, 2, 8, 4, 107, 117, 5952, 15, 256, 4, 2, 7, 3766, 5, 723, 36, 71, 43, 530, 476, 26, 400, 317, 46, 7, 4, 2, 1029, 13, 104, 88, 4, 381, 15, 297, 98, 32, 2071, 56, 26, 141, 6, 194, 7486, 18, 4, 226, 22, 21, 134, 476, 26, 480, 5, 144, 30, 5535, 18, 51, 36, 28, 224, 92, 25, 104, 4, 226, 65, 16, 38, 1334, 88, 12, 16, 283, 5, 16, 4472, 113, 103, 32, 15, 16, 5345, 19, 178, 32]

Review is 1

```
[10]: vocab=imdb.get_word_index() # Retrieve the word index file mapping words to
      ↪indices
      print(vocab)
```

```
{'fawn': 34701, 'tsukino': 52006, 'nunnery': 52007, 'sonja': 16816, 'vani':
63951, 'woods': 1408, 'spiders': 16115, 'hanging': 2345, 'woody': 2289,
'trawling': 52008, 'hold's': 52009, 'comically': 11307, 'localized': 40830,
'disobeying': 30568, 'royale': 52010, 'harpo's': 40831, 'canet': 52011,
'aileen': 19313, 'acurately': 52012, 'diplomat's': 52013, 'rickman': 25242,
'arranged': 6746, 'rumbustious': 52014, 'familiarness': 52015, 'spider': 52016,
'hahahah': 68804, 'wood': 52017, 'transvestism': 40833, 'hangin': 34702,
'bringing': 2338, 'seamier': 40834, 'wooded': 34703, 'bravora': 52018,
'grueling': 16817, 'wooden': 1636, 'wednesday': 16818, 'prix': 52019,
'altagracia': 34704, 'circuitry': 52020, 'crotch': 11585, 'busybody': 57766,
'tart'n'tangy': 52021, 'burgade': 14129, 'thrace': 52023, 'tom's': 11038,
'snuggles': 52025, 'francesco': 29114, 'complainers': 52027, 'templarios':
52125, '272': 40835, '273': 52028, 'zaniacs': 52130, '275': 34706, 'consenting':
27631, 'snuggled': 40836, 'inanimate': 15492, 'uality': 52030, 'bronte': 11926,
'errors': 4010, 'dialogs': 3230, 'yomada's': 52031, 'madman's': 34707,
'dialoge': 30585, 'usenet': 52033, 'videodrome': 40837, 'kid': 26338, 'pawed':
52034, 'girlfriend': 30569, 'pleasure': 52035, 'reloaded': 52036,
'kazakos': 40839, 'rocque': 52037, 'mailings': 52038, 'brainwashed': 11927,
'mcanally': 16819, 'tom': 52039, 'kurupt': 25243, 'affiliated': 21905,
'babaganoosh': 52040, 'noe's': 40840, 'quart': 40841, 'kids': 359, 'uplifting':
5034, 'controversy': 7093, 'kida': 21906, 'kidd': 23379, 'error': 52041,
'neurologist': 52042, 'spotty': 18510, 'cobblers': 30570, 'projection': 9878,
'fastforwarding': 40842, 'sters': 52043, 'eggars': 52044, 'etherything': 52045,
'gateshead': 40843, 'airball': 34708, 'unsinkable': 25244, 'stern': 7180,
'cervi's': 52046, 'dnd': 40844, 'dna': 11586, 'insecurity': 20598, 'reboot':
52047, 'trelkovsky': 11037, 'jaekel': 52048, 'sidebars': 52049, 'sforza's':
52050, 'distortions': 17633, 'mutinies': 52051, 'sermons': 30602, '7ft': 40846,
'boobage': 52052, 'o'bannon's': 52053, 'populations': 23380, 'chulak': 52054,
'mesmerize': 27633, 'quinnell': 52055, 'yahoo': 10307, 'meteorologist': 52057,
'beswick': 42577, 'boorman': 15493, 'voicework': 40847, 'ster': 52058,
```

'blustering': 22922, 'hj': 52059, 'intake': 27634, 'morally': 5621, 'jumbling':  
 40849, 'bowersock': 52060, "'porky's'": 52061, 'gershon': 16821, 'ludicrously':  
 40850, 'coprophilia': 52062, 'expressively': 40851, "india's": 19500, "post's":  
 34710, 'wana': 52063, 'wang': 5283, 'wand': 30571, 'wane': 25245, 'edgeways':  
 52321, 'titanium': 34711, 'pinta': 40852, 'want': 178, 'pinto': 30572,  
 'whoopeddoodles': 52065, 'tchaikovsky': 21908, 'travel': 2103, "'victory'":  
 52066, 'copious': 11928, 'gouge': 22433, "chapters'": 52067, 'barbra': 6702,  
 'uselessness': 30573, "wan'": 52068, 'assimilated': 27635, 'petiot': 16116,  
 'most\x85and': 52069, 'dinosaurs': 3930, 'wrong': 352, 'seda': 52070, 'stollen':  
 52071, 'sentencing': 34712, 'ouroboros': 40853, 'assimilates': 40854,  
 'colorfully': 40855, 'glenne': 27636, 'dongen': 52072, 'subplots': 4760,  
 'kiloton': 52073, 'chandon': 23381, "effect'": 34713, 'snugly': 27637, 'kuei':  
 40856, 'welcomed': 9092, 'dishonor': 30071, 'concurrence': 52075, 'stoicism':  
 23382, "guys'": 14896, "beroemd'": 52077, 'butcher': 6703, "melfi's": 40857,  
 'aargh': 30623, 'playhouse': 20599, 'wickedly': 11308, 'fit': 1180, 'labratory':  
 52078, 'lifeline': 40859, 'screaming': 1927, 'fix': 4287, 'cineliterate': 52079,  
 'fic': 52080, 'fia': 52081, 'fig': 34714, 'fmvs': 52082, 'fie': 52083,  
 'reentered': 52084, 'fin': 30574, 'doctresses': 52085, 'fil': 52086, 'zucker':  
 12606, 'ached': 31931, 'counsel': 52088, 'paterfamilias': 52089, 'songwriter':  
 13885, 'shivam': 34715, 'hurting': 9654, 'effects': 299, 'slaughter': 52090,  
 "'flame'": 52091, 'sommerset': 52092, 'interwhined': 52093, 'whacking': 27638,  
 'bartok': 52094, 'barton': 8775, 'fewer': 21909, "fi'": 52095, 'ingrid': 6192,  
 'stribor': 30575, 'appropriately': 52096, 'wobblyhand': 52097, 'tantalisingly':  
 52098, 'ankylosaurus': 52099, 'parasites': 17634, 'children': 52100, "jenkins'":  
 52101, 'metafiction': 52102, 'golem': 17635, 'indiscretion': 40860, "reeves'":  
 23383, "inamorata's": 57781, 'britannica': 52104, 'adapt': 7916, "russo's":  
 30576, 'guitarists': 48246, 'abbott': 10553, 'abbots': 40861, 'lanisha': 17649,  
 'magickal': 40863, 'mattter': 52105, "'willy'": 52106, 'pumpkins': 34716,  
 'stuntpeople': 52107, 'estimate': 30577, 'ugghhh': 40864, 'gameplay': 11309,  
 "wern't": 52108, "n'sync": 40865, 'sickeningly': 16117, 'chiara': 40866,  
 'disturbed': 4011, 'portmanteau': 40867, 'ineffectively': 52109, "duchonvey's":  
 82143, "nasty'": 37519, 'purpose': 1285, 'lazars': 52112, 'lightened': 28105,  
 'kaliganj': 52113, 'popularism': 52114, "damme's": 18511, 'stylistics': 30578,  
 'mindgaming': 52115, 'spoilerish': 46449, "'corny'": 52117, 'boerner': 34718,  
 'olds': 6792, 'bakelite': 52118, 'renovated': 27639, 'forrester': 27640,  
 'lumiere's': 52119, 'gaskets': 52024, 'needed': 884, 'smight': 34719, 'master':  
 1297, "edie's": 25905, 'seeber': 40868, 'hiya': 52120, 'fuzziness': 52121,  
 'genesis': 14897, 'rewards': 12607, 'enthrall': 30579, "'about'": 40869,  
 'recollection's': 52122, 'mutilated': 11039, 'fatherlands': 52123, "fischer's":  
 52124, 'positively': 5399, '270': 34705, 'ahmed': 34720, 'zatoichi': 9836,  
 'bannister': 13886, 'anniversaries': 52127, "helm's": 30580, "'work'": 52128,  
 'exclaimed': 34721, "'unfunny'": 52129, '274': 52029, 'feeling': 544, "wanda's":  
 52131, 'dolan': 33266, '278': 52133, 'peacoat': 52134, 'brawny': 40870,  
 'mishra': 40871, 'worlders': 40872, 'protags': 52135, 'skullcap': 52136,  
 'dastagir': 57596, 'affairs': 5622, 'wholesome': 7799, 'hymen': 52137,  
 'paramedics': 25246, 'unpersons': 52138, 'heavyarms': 52139, 'affaire': 52140,  
 'coulisses': 52141, 'hymer': 40873, 'kremlin': 52142, 'shipments': 30581,  
 'pixilated': 52143, "'00s'": 30582, 'diminishing': 18512, 'cinematic': 1357,

```
[11]: y_train
```

```
[11]: array([1, 0, 0, ..., 0, 1, 0], dtype=int64)
```

```
[12]: y_test
```

```
[12]: array([0, 1, 1, ..., 0, 0, 0], dtype=int64)
```

```
[13]: def vectorize(sequences, dimension=10000):  
    results = np.zeros((len(sequences), dimension))  
    for i, sequence in enumerate(sequences):  
        results[i, sequence] = 1  
    return results  
  
test_x = data[:10000]  
test_y = label[:10000]  
train_x = data[10000:]  
train_y = label[10000:]
```

```
[14]: test_x.shape
```

```
[14]: (10000,)
```

```
[15]: test_y.shape
```

```
[15]: (10000,)
```

```
[16]: train_x.shape
```

```
[16]: (40000,)
```

```
[17]: train_y.shape
```

```
[17]: (40000,)
```

```
[18]: print("Categories:", np.unique(label))  
print("Number of unique words:", len(np.unique(np.hstack(data))))
```

```
Categories: [0 1]
```

```
Number of unique words: 9998
```

```
[19]: length = [len(i) for i in data]  
print("Average Review length:", np.mean(length))  
print("Standard Deviation:", round(np.std(length)))
```

```
Average Review length: 234.75892
```

```
Standard Deviation: 173
```

```
[20]: print("Label:", label[0])
```

Label: 1

```
[21]: print("Label:", label[1])
```

Label: 0

```
[22]: print(data[0])
```

```
[1, 14, 22, 16, 43, 530, 973, 1622, 1385, 65, 458, 4468, 66, 3941, 4, 173, 36,
256, 5, 25, 100, 43, 838, 112, 50, 670, 2, 9, 35, 480, 284, 5, 150, 4, 172, 112,
167, 2, 336, 385, 39, 4, 172, 4536, 1111, 17, 546, 38, 13, 447, 4, 192, 50, 16,
6, 147, 2025, 19, 14, 22, 4, 1920, 4613, 469, 4, 22, 71, 87, 12, 16, 43, 530,
38, 76, 15, 13, 1247, 4, 22, 17, 515, 17, 12, 16, 626, 18, 2, 5, 62, 386, 12, 8,
316, 8, 106, 5, 4, 2223, 5244, 16, 480, 66, 3785, 33, 4, 130, 12, 16, 38, 619,
5, 25, 124, 51, 36, 135, 48, 25, 1415, 33, 6, 22, 12, 215, 28, 77, 52, 5, 14,
407, 16, 82, 2, 8, 4, 107, 117, 5952, 15, 256, 4, 2, 7, 3766, 5, 723, 36, 71,
43, 530, 476, 26, 400, 317, 46, 7, 4, 2, 1029, 13, 104, 88, 4, 381, 15, 297, 98,
32, 2071, 56, 26, 141, 6, 194, 7486, 18, 4, 226, 22, 21, 134, 476, 26, 480, 5,
144, 30, 5535, 18, 51, 36, 28, 224, 92, 25, 104, 4, 226, 65, 16, 38, 1334, 88,
12, 16, 283, 5, 16, 4472, 113, 103, 32, 15, 16, 5345, 19, 178, 32]
```

```
[23]: index = imdb.get_word_index()
```

```
[24]: reverse_index = dict([(value, key) for (key, value) in index.items()])
      decoded = " ".join( [reverse_index.get(i - 3, "#") for i in data[0]] )
```

```
[25]: print(decoded)
```

```
# this film was just brilliant casting location scenery story direction
everyone's really suited the part they played and you could just imagine being
there robert # is an amazing actor and now the same being director # father came
from the same scottish island as myself so i loved the fact there was a real
connection with this film the witty remarks throughout the film were great it
was just brilliant so much that i bought the film as soon as it was released for
# and would recommend it to everyone to watch and the fly fishing was amazing
really cried at the end it was so sad and you know what they say if you cry at a
film it must have been good and this definitely was also # to the two little
boy's that played the # of norman and paul they were just brilliant children are
often left out of the # list i think because the stars that play them all grown
up are such a big profile for the whole film but these children are amazing and
should be praised for what they have done don't you think the whole story was so
lovely because it was true and was someone's life after all that was shared with
us all
```

```
[26]: pip install seaborn
```

```

Requirement already satisfied: seaborn in
c:\users\harsh\documents\anaconda\lib\site-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in
c:\users\harsh\documents\anaconda\lib\site-packages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in
c:\users\harsh\documents\anaconda\lib\site-packages (from seaborn) (2.0.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in
c:\users\harsh\documents\anaconda\lib\site-packages (from seaborn) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\harsh\documents\anaconda\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (1.0.5)
Requirement already satisfied: cycler>=0.10 in
c:\users\harsh\documents\anaconda\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\harsh\documents\anaconda\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\harsh\documents\anaconda\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\harsh\documents\anaconda\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\harsh\documents\anaconda\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in
c:\users\harsh\documents\anaconda\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\harsh\documents\anaconda\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\users\harsh\documents\anaconda\lib\site-packages (from pandas>=0.25->seaborn)
(2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\users\harsh\documents\anaconda\lib\site-packages (from pandas>=0.25->seaborn)
(2023.3)
Requirement already satisfied: six>=1.5 in
c:\users\harsh\documents\anaconda\lib\site-packages (from python-
dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

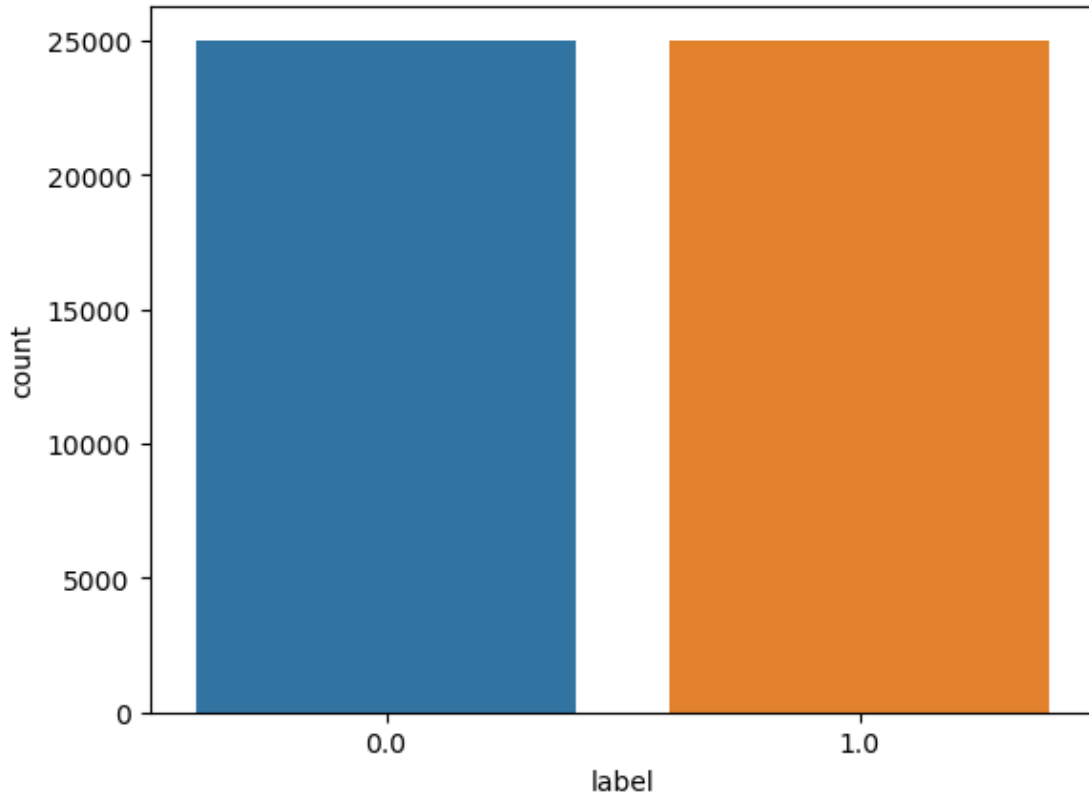
```

```
[27]: import seaborn as sns
```

```
[28]: data = vectorize(data)
      label = np.array(label).astype("float32")
```

```
labelDF = pd.DataFrame({'label': label})
sns.countplot(x='label', data=labelDF)
```

[28]: <Axes: xlabel='label', ylabel='count'>



```
[29]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data, label, test_size=0.20,
    random_state=1)
```

```
[30]: X_train.shape
```

[30]: (40000, 10000)

```
[31]: X_test.shape
```

[31]: (10000, 10000)

```
[32]: from keras.utils import to_categorical
from keras import models
from keras import layers
model = models.Sequential()
```

WARNING:tensorflow:From C:\Users\harsh\Documents\anaconda\Lib\site-packages\keras\src\backend.py:873: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

```
[33]: model.add(layers.Dense(50, activation = "relu", input_shape=(10000, )))
model.add(layers.Dropout(0.3, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation = "relu"))
model.add(layers.Dropout(0.2, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation = "relu"))
model.add(layers.Dense(1, activation = "sigmoid"))
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 50)	500050
dropout (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 50)	2550
dropout_1 (Dropout)	(None, 50)	0
dense_2 (Dense)	(None, 50)	2550
dense_3 (Dense)	(None, 1)	51

=====  
Total params: 505201 (1.93 MB)  
Trainable params: 505201 (1.93 MB)  
Non-trainable params: 0 (0.00 Byte)  
=====

```
[34]: import tensorflow as tf
callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)
model.compile(
    optimizer = "adam",
    loss = "binary_crossentropy",
    metrics = ["accuracy"]
)
from sklearn.model_selection import train_test_split
results = model.fit(
    X_train, y_train,
    epochs= 2,
    batch_size = 500,
```



```

validation_data = (X_test, y_test),
callbacks=[callback]
)
print(np.mean(results.history["val_accuracy"]))
score = model.evaluate(X_test, y_test, batch_size=500)
print('Test loss:', score[0])
print('Test accuracy:', score[1])

```

WARNING:tensorflow:From C:\Users\harsh\Documents\anaconda\Lib\site-packages\keras\src\optimizers\\_\_init\_\_.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

Epoch 1/2

WARNING:tensorflow:From C:\Users\harsh\Documents\anaconda\Lib\site-packages\keras\src\utils\tf\_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\harsh\Documents\anaconda\Lib\site-packages\keras\src\engine\base\_layer\_utils.py:384: The name tf.executing\_eagerly\_outside\_functions is deprecated. Please use tf.compat.v1.executing\_eagerly\_outside\_functions instead.

80/80 [=====] - 7s 58ms/step - loss: 0.4001 - accuracy: 0.8234 - val\_loss: 0.2568 - val\_accuracy: 0.8973

Epoch 2/2

80/80 [=====] - 2s 22ms/step - loss: 0.2146 - accuracy: 0.9170 - val\_loss: 0.2524 - val\_accuracy: 0.8978

0.8975500166416168

20/20 [=====] - 0s 9ms/step - loss: 0.2524 - accuracy: 0.8978

Test loss: 0.2523631751537323

Test accuracy: 0.8978000283241272

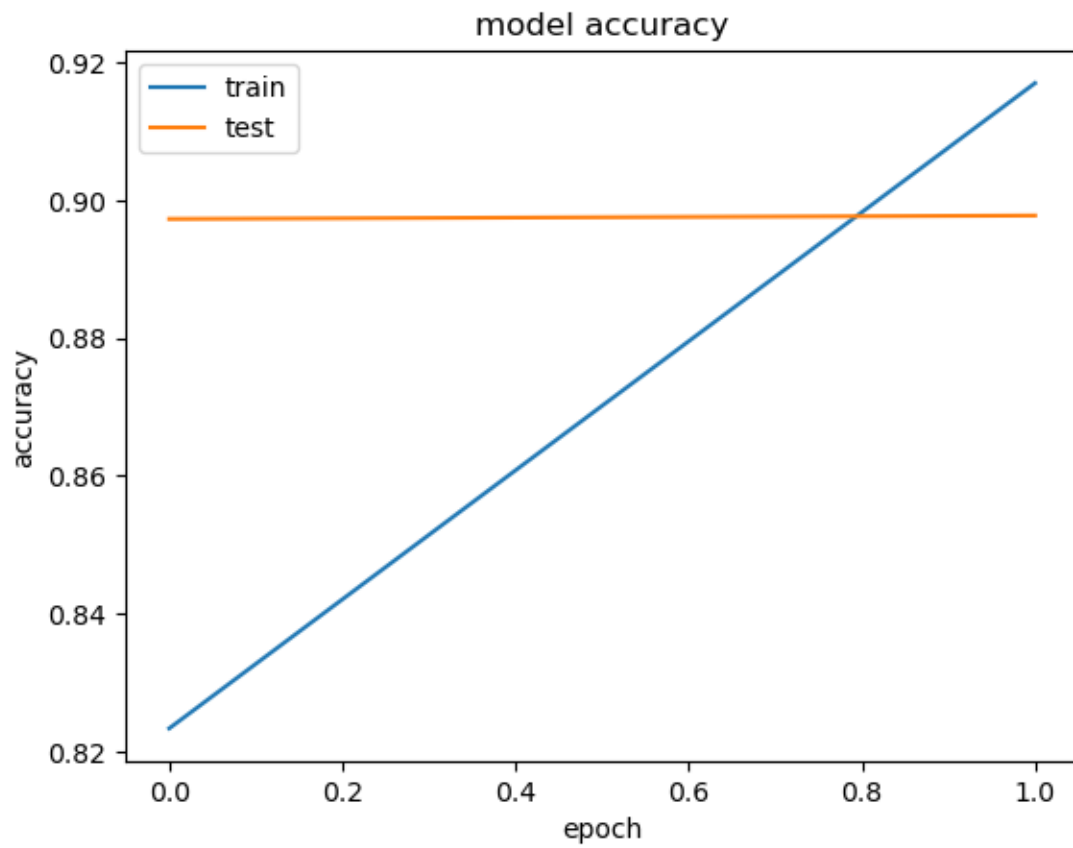
```

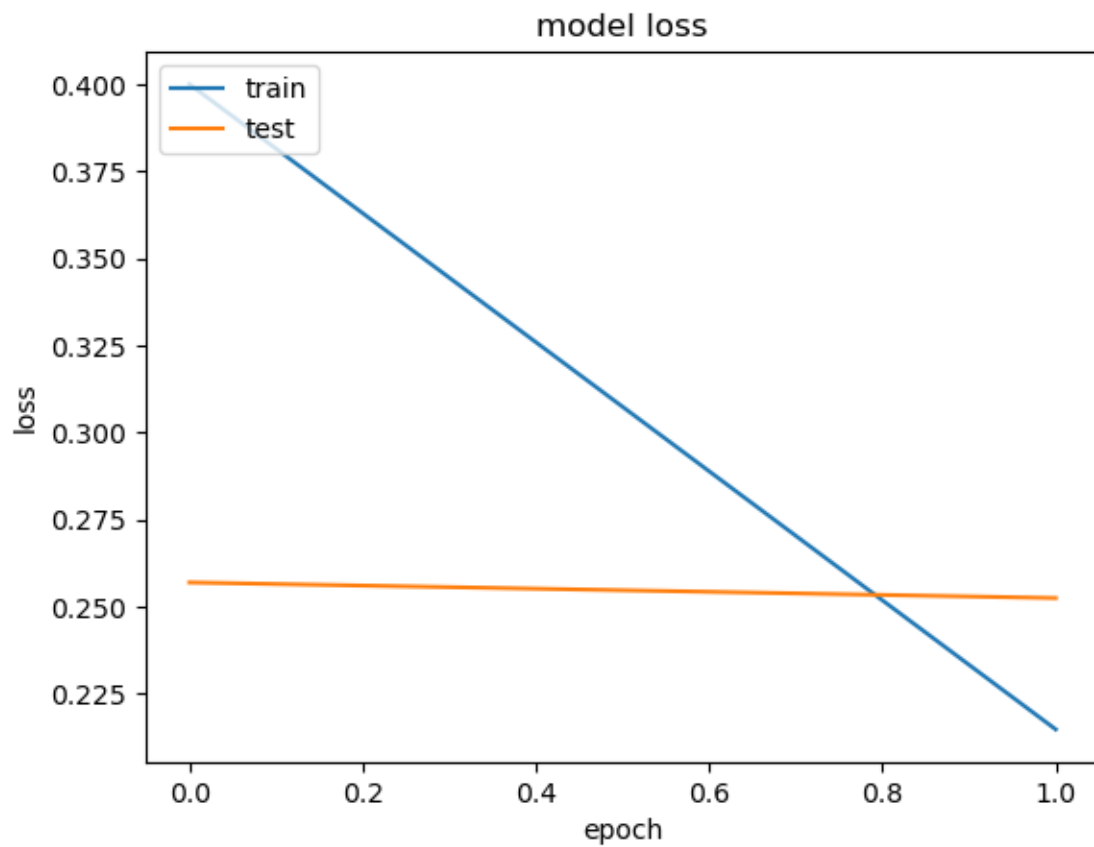
[35]: print(results.history.keys())
plt.plot(results.history['accuracy'])
plt.plot(results.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.plot(results.history['loss'])
plt.plot(results.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')

```

```
plt.show()
```

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```





[ ]: