# dl_2

April 11, 2024

[2]:
```
import numpy as np
import tensorflow as tf
from keras.datasets import imdb
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Embedding, Bidirectional, LSTM, Dense
```

```
C:\Users\Nimisha jadhav\anaconda3\lib\site-packages\scipy\__init__.py:155:
UserWarning: A NumPy version >=1.18.5 and <1.25.0 is required for this version
of SciPy (detected version 1.26.4
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

[3]:
```
# Load the IMDB dataset
(x_train, y_train), (x_test, y_test) = imdb.load_data()
```

[4]:
```
max_len = 250
x_train= pad_sequences(x_train, maxlen=max_len)
x_test = pad_sequences(x_test, maxlen=max_len)
```

[5]:
```
# Define the deep neural network architecture
model = Sequential()
model.add(Embedding(input_dim=10000, output_dim=128, input_length=max_len))
model.add(Bidirectional(LSTM(64, return_sequences=True)))
model.add(Bidirectional(LSTM(32)))
model.add(Dense(1, activation='sigmoid'))
```

```
C:\Users\Nimisha jadhav\anaconda3\lib\site-
packages\keras\src\layers\core\embedding.py:86: UserWarning: Argument
`input_length` is deprecated. Just remove it.
  warnings.warn(
```

[6]:
```
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])
```

[7]:
```
print(x_train.min(), x_train.max())
```

```
0 88586
```

[9]:
```
history = model.fit(x_train, y_train, epochs=10, batch_size=128,␣
↪validation_split=0.2)
```

```
Epoch 1/10
157/157             254s 2s/step -
accuracy: 0.6688 - loss: 0.5786 - val_accuracy: 0.8624 - val_loss: 0.3332
Epoch 2/10
157/157             1413s 9s/step -
accuracy: 0.9077 - loss: 0.2435 - val_accuracy: 0.8544 - val_loss: 0.3348
Epoch 3/10
157/157             188s 1s/step -
accuracy: 0.9448 - loss: 0.1603 - val_accuracy: 0.8794 - val_loss: 0.3341
Epoch 4/10
157/157             186s 1s/step -
accuracy: 0.9663 - loss: 0.1071 - val_accuracy: 0.8492 - val_loss: 0.4181
Epoch 5/10
157/157             185s 1s/step -
accuracy: 0.9735 - loss: 0.0849 - val_accuracy: 0.8274 - val_loss: 0.4931
Epoch 6/10
157/157             192s 1s/step -
accuracy: 0.9680 - loss: 0.0961 - val_accuracy: 0.8612 - val_loss: 0.4649
Epoch 7/10
157/157             190s 1s/step -
accuracy: 0.9854 - loss: 0.0529 - val_accuracy: 0.8210 - val_loss: 0.6263
Epoch 8/10
157/157             192s 1s/step -
accuracy: 0.9822 - loss: 0.0580 - val_accuracy: 0.8600 - val_loss: 0.5804
Epoch 9/10
157/157             193s 1s/step -
accuracy: 0.9886 - loss: 0.0373 - val_accuracy: 0.8596 - val_loss: 0.5646
Epoch 10/10
157/157             205s 1s/step -
accuracy: 0.9916 - loss: 0.0319 - val_accuracy: 0.8598 - val_loss: 0.6139
```

[11]:
```
# Evaluate the model on the test setloss
loss, acc = model.evaluate(x_test, y_test, batch_size=128)
print(f'Test accuracy: {acc:.4f}, Test loss: {loss:.4f}')
```

```
196/196             56s 283ms/step -
accuracy: 0.8555 - loss: 0.6368
Test accuracy: 0.8558, Test loss: 0.6379
```

[ ]:
```

```