

PS11

April 14, 2023

0.1 Recurrent Neural Networks

```
[1]: import tensorflow as tf
      from tensorflow.keras.datasets import imdb
      import numpy as np
```

0.2 Loading IMDB

```
[2]: from tensorflow.keras.datasets import imdb

      (X_train,y_train), (X_test,y_test) = imdb.load_data(num_words=10000)
      # (X_train,y_train), (X_test,y_test) = imdb.load_data(num_words=500)
```

```
[3]: # Get IMDB dictionary
      word_idx_dict = imdb.get_word_index()

      #print(word_idx_dict.keys())
      #print(word_idx_dict.values())
```

0.3 Conversion from numerical values back to words

0.3.1 Conversion without taking “+3” into consideration

```
[4]: reverse_word_idx_dict_ex = {}
      for key, value in word_idx_dict.items():
          reverse_word_idx_dict_ex[value]=key

[5]: def decode_review_ex(vector):
      word_list=[]
      for value in vector:
          word_list.append(reverse_word_idx_dict_ex[value])
      return ' '.join(word_list)

[6]: sample = decode_review_ex(X_train[0])
      print(len(X_train[0]))
      print(sample)
```

218

the as you with out themselves powerful lets loves their becomes reaching had
journalist of lot from anyone to have after out atmosphere never more room and
it so heart shows to years of every never going and help moments or of every
chest visual movie except her was several of enough more with is now current
film as you of mine potentially unfortunately of you than him that with out
themselves her get for was camp of you movie sometimes movie that with scary but
and to story wonderful that in seeing in character to of 70s musicians with
heart had shadows they of here that with her serious to have does when from why
what have critics they is you that isn't one will very to as itself with other
and in of seen over landed for anyone of and br show's to whether from than out
themselves history he name half some br of and odd was two most of mean for 1
any an boat she he should is thought frog but of script you not while history he
heart to real at barrel but when from one bit then have two of script their with
her nobody most that with wasn't to with armed acting watch an for with
heartfelt film want an

0.3.2 Conversion with taking “+3” into consideration

```
[7]: # The first three values are assigned for controlling a sentence
word_idx_dict_new={}
for key, value in word_idx_dict.items():
    word_idx_dict_new[key]=value+3
```

```
[8]: print(word_idx_dict['the'])
      print(word_idx_dict_new['the'])
```

1
4

```
[9]: word_idx_dict_new['<PAD>']=0
      word_idx_dict_new['<START>']=1
      word_idx_dict_new['<UNK>']=2
      word_idx_dict_new['<UNUSED>']=3
```

```
#word_idx_dict_new
```

```
[10]: reverse_word_idx_dict={}
      for key, value in word_idx_dict_new.items():
          reverse_word_idx_dict[value]=key
```

```
[11]: print(reverse_word_idx_dict[0])
      print(reverse_word_idx_dict[1])
      print(reverse_word_idx_dict[2])
      print(reverse_word_idx_dict[3])
```

<PAD>
<START>

<UNK>
<UNUSED>

```
[12]: def decode_review(vector):  
      word_list=[]  
      for value in vector:  
          word_list.append(reverse_word_idx_dict[value])  
      return ' '.join(word_list)
```

```
[13]: print(decode_review(X_train[0]))
```

<START> 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 <UNK> is an amazing actor and now the same being director <UNK> 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 <UNK> 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 <UNK> to the two little boy's that played the <UNK> of norman and paul they were just brilliant children are often left out of the <UNK> 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

```
[14]: print(y_train[0])
```

1

0.4 Data preprocessing: Padding

```
[15]: from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
[16]: print(len(X_train[0]))
```

218

```
[17]: max_len = 256  
      X_train_pad = pad_sequences(X_train,value=0, padding='post', maxlen=256)
```

```
[18]: print(X_train_pad.shape)
```

(25000, 256)

```
[19]: print(decode_review(X_train_pad[0]))
```

<START> 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 <UNK> is an amazing actor and now the same being director <UNK> 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 <UNK> 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 <UNK> to the two little boy's that played the <UNK> of norman and paul they were just brilliant children are often left out of the <UNK> 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 <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
<PAD> <PAD> <PAD> <PAD> <PAD> <PAD>

```
[20]: X_test_pad = pad_sequences(X_test, value=0, padding='post', maxlen=max_len)
```

```
[21]: print(X_train_pad.shape)
      print(X_test_pad.shape)
```

(25000, 256)

(25000, 256)

0.5 Word embedding

```
[22]: from tensorflow.keras.models import Sequential
      from tensorflow.keras.layers import Dense
      from tensorflow.keras.layers import Embedding

      model = Sequential()
      model.add(Embedding(input_dim=10000, output_dim=100, input_length=256))
      model.summary()

      #model.add(Dense(units=128, activation='relu'))
      #model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 256, 100)	1000000

```
=====
Total params: 1,000,000
Trainable params: 1,000,000
Non-trainable params: 0
-----
```

[]: