RNN_DeepL

November 8, 2023

Estefania Pérez Yeo - A01639270

---

1 La Biblia en Inglés

[2]:
```
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
import os
```

2 Descarga de datos

[3]:
```
path_to_fileDL = tf.keras.utils.get_file('bible.txt', 'https://raw.
↪githubusercontent.com/mxw/grmr/master/src/finaltests/bible.txt')

text = open (path_to_fileDL, 'rb').read().decode(encoding='utf-8')
print('Longitud del texto: {} caracteres'.format(len(text)))

vocab = sorted(set(text))
print('El texto está compuesto de estos {} caracteres'.format(len(vocab)))
print(vocab)
```

```
Downloading data from
https://raw.githubusercontent.com/mxw/grmr/master/src/finaltests/bible.txt
4451368/4451368 [==============================] - 0s 0us/step
Longitud del texto: 4451368 caracteres
El texto está compuesto de estos 81 caracteres
['\n', '\r', ' ', '!', '"', '$', '%', "'", '(', ')', '*', ',', '-', '.', '/',
'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', ':', ';', '?', '@', 'A', 'B',
'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R',
'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
'y', 'z']
```

3 Tablas de traduccion o iversa de vocabulario

[4]:
```
char2idx = {u:i for i, u in enumerate(vocab)}
idx2char = np.array(vocab)
```

[5]:
```
for char,_ in zip(char2idx, range(len(vocab))):
  print(' {:4s}: {:3d},'.format(repr(char), char2idx[char]))
```

```
'\n':   0,
'\r':   1,
' ' :   2,
'!' :   3,
'"' :   4,
'$' :   5,
'%' :   6,
"'" :   7,
'(' :   8,
')' :   9,
'*' :  10,
',' :  11,
'-' :  12,
'.' :  13,
'/' :  14,
'0' :  15,
'1' :  16,
'2' :  17,
'3' :  18,
'4' :  19,
'5' :  20,
'6' :  21,
'7' :  22,
'8' :  23,
'9' :  24,
':' :  25,
';' :  26,
'?' :  27,
'@' :  28,
'A' :  29,
'B' :  30,
'C' :  31,
'D' :  32,
'E' :  33,
'F' :  34,
'G' :  35,
'H' :  36,
'I' :  37,
'J' :  38,
'K' :  39,
```

```
'L' :  40,
'M' :  41,
'N' :  42,
'O' :  43,
'P' :  44,
'Q' :  45,
'R' :  46,
'S' :  47,
'T' :  48,
'U' :  49,
'V' :  50,
'W' :  51,
'X' :  52,
'Y' :  53,
'Z' :  54,
'a' :  55,
'b' :  56,
'c' :  57,
'd' :  58,
'e' :  59,
'f' :  60,
'g' :  61,
'h' :  62,
'i' :  63,
'j' :  64,
'k' :  65,
'l' :  66,
'm' :  67,
'n' :  68,
'o' :  69,
'p' :  70,
'q' :  71,
'r' :  72,
's' :  73,
't' :  74,
'u' :  75,
'v' :  76,
'w' :  77,
'x' :  78,
'y' :  79,
'z' :  80,
```

4 Convertir texto a enteros

[6]:
```
text_as_int = np.array([char2idx[c] for c in text])
```

[7]:
```
#Mostramos algunos caracteres
print('text: {}'.format(repr(text[:50])))
print('{}'.format(repr(text_as_int[:50])))
```

```
text: '1:1 In the beginning God created the heaven and th'
array([16, 25, 16,  2, 37, 68,  2, 74, 62, 59,  2, 56, 59, 61, 63, 68, 68,
       63, 68, 61,  2, 35, 69, 58,  2, 57, 72, 59, 55, 74, 59, 58,  2, 74,
       62, 59,  2, 62, 59, 55, 76, 59, 68,  2, 55, 68, 58,  2, 74, 62])
```

5 Preparar datos

[8]:
```
char_dataset = tf.data.Dataset.from_tensor_slices(text_as_int)
seq_length = 100
sequences = char_dataset.batch(seq_length+1, drop_remainder=True)
```

[9]:
```
#comprobar datos
for item in sequences.take(10):
  print(repr(''.join(idx2char[item.numpy()])))
```

```
'1:1 In the beginning God created the heaven and the earth.\r\n\r\n1:2 And the
earth was without form, and'
' void; and darkness was upon\r\nthe face of the deep. And the Spirit of God
moved upon the face of the\r'
'\nwaters.\r\n\r\n1:3 And God said, Let there be light: and there was
light.\r\n\r\n1:4 And God saw the light, '
'that it was good: and God divided the light\r\nfrom the darkness.\r\n\r\n1:5
And God called the light Day, '
'and the darkness he called Night.\r\nAnd the evening and the morning were the
first day.\r\n\r\n1:6 And God'
' said, Let there be a firmament in the midst of the waters,\r\nand let it
divide the waters from the wa'
'ters.\r\n\r\n1:7 And God made the firmament, and divided the waters which
were\r\nunder the firmament from '
'the waters which were above the firmament:\r\nand it was so.\r\n\r\n1:8 And God
called the firmament Heaven'
'. And the evening and the\r\nmorning were the second day.\r\n\r\n1:9 And God
said, Let the waters under the'
' heaven be gathered together\r\nunto one place, and let the dry land appear:
and it was so.\r\n\r\n1:10 And'
```

[10]:
```
#Preparar datos de entrenamiento (Entrada 0 a 99 ) (Salida 1 a 100)
def split_input_target(chunk):
  input_text = chunk[:-1]
  target_text = chunk[1:]
  return input_text, target_text
dataset = sequences.map(split_input_target)
```

[11]:
```
#Visualizamos
for input_example, target_example in dataset.take(1):
  print ('Input data: ', repr(''.join(idx2char[input_example.numpy()])))
  print ('Target data: ', repr(''.join(idx2char[target_example.numpy()])))
```

```
Input data:  '1:1 In the beginning God created the heaven and the
earth.\r\n\r\n1:2 And the earth was without form, an'
Target data:  ':1 In the beginning God created the heaven and the
earth.\r\n\r\n1:2 And the earth was without form, and'
```

[12]:
```
#imprimir dataset
print (dataset)
```

```
<_MapDataset element_spec=(TensorSpec(shape=(100,), dtype=tf.int64, name=None),
TensorSpec(shape=(100,), dtype=tf.int64, name=None))>
```

[13]:
```
#agrupar en batches
BATCH_SIZE = 64
BUFFER_SIZE = 10000
dataset = dataset.shuffle(BUFFER_SIZE).batch(BATCH_SIZE, drop_remainder=True)
print(dataset)
```

```
<_BatchDataset element_spec=(TensorSpec(shape=(64, 100), dtype=tf.int64,
name=None), TensorSpec(shape=(64, 100), dtype=tf.int64, name=None))>
```

6 Construir modelo RNN

[14]:
```
def build_model(vocab_size, embedding_dim, rnn_units, batch_size):
  model = tf.keras.Sequential([
  tf.keras.layers.Embedding(vocab_size, embedding_dim,
  batch_input_shape=[batch_size,None]),
  tf.keras.layers.LSTM(rnn_units,
  return_sequences=True,
  stateful = True,
  recurrent_initializer='glorot_uniform'),
  tf.keras.layers.Dense(vocab_size)
  ])
  return model
```

[15]:
```
vocab_size = len(vocab)
embedding_dim= 256
rnn_units = 1024
```

[16]:
```
model = build_model(
  vocab_size = vocab_size,
  embedding_dim=embedding_dim,
  rnn_units=rnn_units,
  batch_size = BATCH_SIZE
```

```
)
```

[17]:
```
#Visualizar estructura
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (64, None, 256)           20736

 lstm (LSTM)                 (64, None, 1024)          5246976

 dense (Dense)               (64, None, 81)            83025

=================================================================
Total params: 5350737 (20.41 MB)
Trainable params: 5350737 (20.41 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[18]:
```
# Forma de input
for input_example_batch, target_example_batch in dataset.take(1):
  print("Input: ", input_example_batch.shape, "# (batch_size, lenght)")
  print("Target: ", target_example_batch.shape, "# (batch_size,␣
 ↪sequence_length)")
```

```
Input:  (64, 100) # (batch_size, lenght)
Target:  (64, 100) # (batch_size, sequence_length)
```

[19]:
```
#Forma de salida
for input_example_batch, target_example_batch in dataset.take(1):
  example_batch_predictions = model(input_example_batch)
  print("Prediction: ", example_batch_predictions.shape, "# (batch_size,␣
 ↪sequence_length, vocab_size)")
```

```
Prediction:  (64, 100, 81) # (batch_size, sequence_length, vocab_size)
```

[20]:
```
#Mostar que el resultado es una distribucion, no un argmax
sampled_indices = tf.random.categorical(example_batch_predictions[0],␣
 ↪num_samples=1)
sampled_indices_characters = tf.squeeze(sampled_indices,axis=-1).numpy()
print(sampled_indices_characters)
```

```
[56 32 28 80 40 32  0 35 66 16 12 11 26 20 78 72 30 43 74 51 18 55 69 31
  6 31 76 79 76 10 60  7 20 77 73 72 47 68 57 68 55 56 74 33 29 21 42 39
 40 74 65 71 51 66 13 10 38 10 37 20  9  7 37 56 61 40 64 36 73 39 32 28
 71 22 16 64 14 11 59 65 24 42  3  3 19 21 11 77 19 16 78 47 37 16 36 79
 18 60 78 60]
```

7 Entrenamiento

[21]:
```
def loss(labels, logits):
  return tf.keras.losses.sparse_categorical_crossentropy(labels, logits,␣
 ↪from_logits=True)
```

[22]:
```
model.compile(optimizer='adam', loss=loss)
```

[23]:
```
checkpoint_dir = './training_checkpoints'
checkpoint_prefix = os.path.join(checkpoint_dir, "ckpt_(epoch)")
checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(
  filepath=checkpoint_prefix,
  save_weights_only=True
)
```

[24]:
```
EPOCHS = 50
history = model.fit(dataset, epochs=EPOCHS, callbacks=[checkpoint_callback])
```

```
Epoch 1/50
688/688 [==============================] - 51s 68ms/step - loss: 1.6503
Epoch 2/50
688/688 [==============================] - 50s 70ms/step - loss: 1.1439
Epoch 3/50
688/688 [==============================] - 50s 70ms/step - loss: 1.0614
Epoch 4/50
688/688 [==============================] - 49s 70ms/step - loss: 1.0169
Epoch 5/50
688/688 [==============================] - 50s 70ms/step - loss: 0.9845
Epoch 6/50
688/688 [==============================] - 50s 70ms/step - loss: 0.9585
Epoch 7/50
688/688 [==============================] - 50s 69ms/step - loss: 0.9360
Epoch 8/50
688/688 [==============================] - 50s 70ms/step - loss: 0.9161
Epoch 9/50
688/688 [==============================] - 49s 69ms/step - loss: 0.8980
Epoch 10/50
688/688 [==============================] - 49s 69ms/step - loss: 0.8817
Epoch 11/50
688/688 [==============================] - 49s 69ms/step - loss: 0.8665
Epoch 12/50
688/688 [==============================] - 50s 70ms/step - loss: 0.8525
Epoch 13/50
688/688 [==============================] - 50s 70ms/step - loss: 0.8396
Epoch 14/50
688/688 [==============================] - 50s 71ms/step - loss: 0.8281
Epoch 15/50
688/688 [==============================] - 50s 70ms/step - loss: 0.8174
```

```
Epoch 16/50
688/688 [==============================] - 50s 69ms/step - loss: 0.8087
Epoch 17/50
688/688 [==============================] - 49s 70ms/step - loss: 0.8001
Epoch 18/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7927
Epoch 19/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7861
Epoch 20/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7811
Epoch 21/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7754
Epoch 22/50
688/688 [==============================] - 49s 70ms/step - loss: 0.7716
Epoch 23/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7684
Epoch 24/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7658
Epoch 25/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7627
Epoch 26/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7612
Epoch 27/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7599
Epoch 28/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7597
Epoch 29/50
688/688 [==============================] - 53s 71ms/step - loss: 0.7588
Epoch 30/50
688/688 [==============================] - 51s 70ms/step - loss: 0.7590
Epoch 31/50
688/688 [==============================] - 54s 71ms/step - loss: 0.7590
Epoch 32/50
688/688 [==============================] - 51s 71ms/step - loss: 0.7600
Epoch 33/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7603
Epoch 34/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7620
Epoch 35/50
688/688 [==============================] - 50s 69ms/step - loss: 0.7631
Epoch 36/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7656
Epoch 37/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7678
Epoch 38/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7704
Epoch 39/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7730
```

```
Epoch 40/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7755
Epoch 41/50
688/688 [==============================] - 52s 70ms/step - loss: 0.7787
Epoch 42/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7811
Epoch 43/50
688/688 [==============================] - 49s 69ms/step - loss: 0.7855
Epoch 44/50
688/688 [==============================] - 49s 70ms/step - loss: 0.7911
Epoch 45/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7924
Epoch 46/50
688/688 [==============================] - 50s 70ms/step - loss: 0.7966
Epoch 47/50
688/688 [==============================] - 49s 69ms/step - loss: 0.8015
Epoch 48/50
688/688 [==============================] - 51s 70ms/step - loss: 0.8067
Epoch 49/50
688/688 [==============================] - 50s 70ms/step - loss: 0.8124
Epoch 50/50
688/688 [==============================] - 50s 70ms/step - loss: 0.8173
```

8 Generación de texto

[25]:
```
model = build_model(vocab_size, embedding_dim, rnn_units, batch_size=1)
model.load_weights(tf.train.latest_checkpoint(checkpoint_dir))
model.build(tf.TensorShape([1,None]))
```

[27]:
```
def generate_text(model, start_string,temp):
  num_generate = 500
  input_eval = [char2idx[s] for s in start_string]
  input_eval = tf.expand_dims(input_eval, 0)
  text_generated = []
  temperature = temp
  model.reset_states()
  for i in range(num_generate):
    predictions = model(input_eval)
    predictions = tf.squeeze(predictions,0)
    predictions = predictions/temperature
    predicted_id = tf.random.categorical(predictions, num_samples=1)[-1,0].
↪numpy()
    input_eval = tf.expand_dims([predicted_id],0)
    text_generated.append(idx2char[predicted_id])
  return(start_string + ''.join(text_generated))
```

9 Entrada 1

9.1 Temperatura 1

[29]:
```
print(generate_text(model, start_string=u"God", temp = 0.5))
```

```
God-KYYYYY for the four winds of the chambers, the first of your seas, and the
fruits of
your particulas, that ye may be ashamed of your eyes to do them.

22:24 I am the law of the house of Israel, that they may be
converted by the sword.

6:2 And when the evening angels were come together in the
midst of the sea, they said, He doeth the word of God.

14:20 And after three days, when they were come to him that he was a drunkard,
or
the whoredom, the faith of the hearing of the world, but
```

9.2 Temperatura 2

[30]:
```
print(generate_text(model, start_string=u"God", temp = 1.0))
```

```
God; The Father of the
Jew began to reheard us in the Lord. And there were also a mighty man know
nothing.

1:28 So ou 42:22 Whereupon, thy sons, shall be a village of thy love shall never
take balm with a strong praise
nor swear; 2:2 Shew thyself, that ture Israel concerning this Jews, and
what went they unto them, Go, what shallum I bowed me, I will make
thee anguise from speakers and honour from thy neck, and shall not be poured
out?  7:25 As it
was in the bodd, neither eat they also bri
```

9.3 Temperatura 3

[31]:
```
print(generate_text(model, start_string=u"God", temp = 2.0))
```

```
God-Qut thy
hurtslatepar6:25:37 O bany hunwle was
Phiri15:11 Away, shewing
kingdoms any;) 6376) 50:3 Shall I escaped it; keep
gLORD commandece.

8:53 For if I am little? Pams took it etern! tflick:5 Envip:4 Kay withes, a
pettemen begove
tharaour.
```

```
3 saidext, and up,
whithe frur of
Zaodyeard comy therefore:
forasmuch's wo? ye dose my
kinddom, blotctiKign unwablour be? si Phood
of his glory.

40:12 And imAndistise ye also
uprigns.

13:18 If a wood ool? your Ferst's kills, telvapel
```

10 Entrada 2

10.1 Temperatura 1

[32]:
```
print(generate_text(model, start_string=u"Jesus", temp = 4.0))
```

```
Jesus;148 glo ,.46 Pp,s Suct, aKmoN.

8:10sSchTpi
Lordanacb
Deuniwenle:pl isid7082u186:3 iof
wogh,
VLVR.h
itl5:)2,
Y32H,un ?
hd.E.hDX42.
7 hAd

36:59YVec2: puise? aAUrmiw
CJab"s.
fy, Log d lhe(Blef:26 ButmMow
see V5bal.lvVMai filfyiohaur.
AeLih? Net fivO Epxri:
l,? of, nebe:.7ndascrps. Clmae,
Tofinmut;)2?gfy mag,
xiahai,
n Lms's9 Sgs HEbiSYl uO
*aSue, aCprido(Aod, Ve n
1Vtwt.
88:pa; gier;1d.0 waow day09:7aki mu jud;*kin driact. .7
Tue
```

10.2 Temperature 2

[33]:
```
print(generate_text(model, start_string=u"Jesus",temp = 2.5))
```

```
Jesusly,
64:21 Ye ju:9  19.3

B

T9SLE176.9.158 alp upon Prue glory veyr took Jesub: hf; snu w.*

rox.

29:85 Bubmien liberty;
Thbngy, l),
vexavis,
s, if ley
me sojuplt, let
himpsflumiuty. Of ago site fuge fayes: yeBried
Psiaclpad, zroes cabry than w;9 Repbo to 's fre,  vi ftuch. Ad
L7:28:1:20 Turn ugheki.
Slain Kvirg, ponderiariohdred;28:All be genty en.

7:84"He pst.0:47
Grierey time;
immeth, since LISrae:We quesny. Wremath, frazrifight and duivert: 7:2
7erEch, cryft.oenahgn
```

10.3 Temperature 3

[34]:
```
print(generate_text(model, start_string=u"Jesus", temp = 1.5))
```

```
Jesus disannalled with
any prophet among
you.

4:52 Gide nothousand by a women against whether, and converting them to
ride: if your leay fall, he alwest well in Joppa all likewise.

9:31 And I will take him that mourned: osey the families of
Dai, 28:29 Which which belinved is quite 1:9 He fashion and his power shewiver
athents, peace.,
people:some must spend bread.

57:4 Who  not glory, for our ways.
Forth priests, to horse, and smote it up of lusts,
that I may not understable white.
```