GIT Department of Computer Engineering

CSE 654 / 484 Fall 2022

Homework 4 # Report

Okan Torun

1801042662

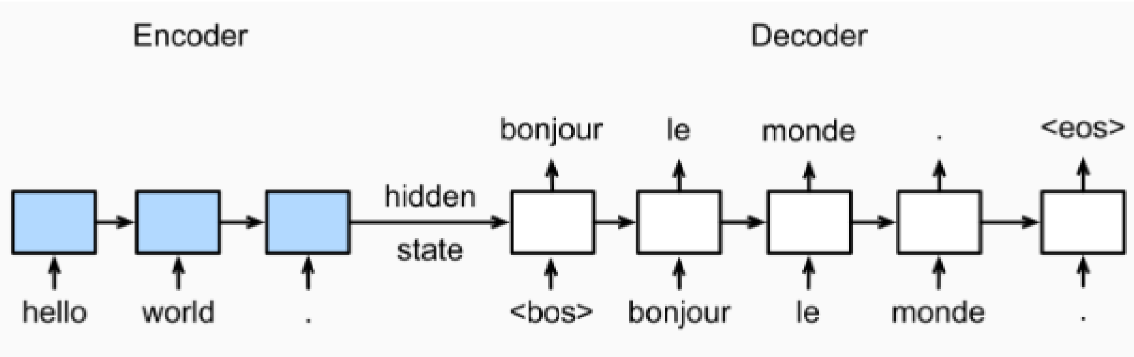
How I Handled the Problem:

The subject of the assignment is about converting sequences in one domain (eg Ottoman sentences) to sequences in another field (eg sentences in Turkish). First of all, a model based on LSTM methods had to be found. I chose the Seq2Seq model as the model selection.

What is Seq2Seq Model?

Sequence-to-sequence learning (Seq2Seq) is about training models for converting sequences in one domain (e.g. Ottoman sentences) into sequences in another domain (e.g. the same sentences translated into Turkish). Our aim is to translate the given sentences from Ottoman to Turkish.

Here both input and output are sentences. In other words, these sentences are a series of words that go in and out of a pattern. This is the basic idea of Array-to-Array modeling. The figure below attempts to explain this method.



To train a Seq2Seq model for this translator assignment, you will need to provide it with a dataset of Ottoman phrases and their corresponding modern-day Turkish translations. The model will then learn to map out Ottoman introductory phrases to output modern Turkish translations.

Method Details:

```
def map_files(file1,file2):
  with open(file1) as f1, open(file2) as f2:
    lines1 = f1.readlines()
    lines2 = f2.readlines()
    merged_lines = [line1.strip() + '*' + line2.strip() for line1, line2 in zip(lines1, lines2)]

  with open('merged_file.txt', 'w') as f:
      for line in merged_lines:
          f.write(line + '\n')
```

First of all, I wanted to map the Ottoman and Turkish datasets in two different files, line by line, to make them suitable for the data set. In this function, I read the two datasets line by line and put a special character (*) between them on each line to reveal the meaning of the line corresponding to a line.

```
def split_lines(text):
    lines = text.strip().split('\n')
    lines = [i.split('*') for i in lines]
    return lines
```

Here, I split the lines I mapped by reading line by line from the merged file that I combined two datasets in common.

```
ottoman_turkish = ottoman_turkish[:30000,:]
```

I specify the number of rows I want to work with. Since model training is very slow, it can be faster to reach the result with small datasets.

```
ottoman_turkish[:,0] = [s.translate(str.maketrans('', '', string.punctuation)) for s in ottoman_turkish[:,0]]
ottoman_turkish[:,1] = [s.translate(str.maketrans('', '', string.punctuation)) for s in ottoman_turkish[:,1]]
```

```
# convert text to lowercase
for i in range(len(ottoman_turkish)):
    ottoman_turkish[i,0] = ottoman_turkish[i,0].lower()
    ottoman_turkish[i,1] = ottoman_turkish[i,1].lower()
```

In order to make the lines I mapped suitable for the model, I remove the punctuation marks and convert them to lowercase. I am performing preprocessing.

```
def tokenization(lines):
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(lines)
    return tokenizer
```

A **Seq2Seq** model allows us to convert both input and output sentences into fixed-length integer sequences.Using Keras's Tokenizer() class, Turkish data and the corresponding Ottoman data are vectorized.

fit_on_texts() uses the tokenizer instance's fit_on_texts() method to create a dictionary index based on the words in the input lines. The token generator creates a match between each word in the dictionary and a unique integer value.

```
turkish_tokenizer = tokenization(ottoman_turkish[:, 0])
turkish_vocab_size = len(turkish_tokenizer.word_index) + 1

turkish_length = 8
print('Turkish Vocabulary Size: %d' % turkish_vocab_size)
```

```
Turkish Vocabulary Size: 49464
```

The first column of the "ottoman_turkish" array contains a list of Turkish phrases. The tokenization function creates a word index based on the words in these Turkish expressions.

The next line defines a variable named "Turkish_vocab_size" where "turkish_tokenizer" plus 1 is the length of the "word_index" attribute. The "word_index" attribute is a dictionary that maps each word in the dictionary to its corresponding integer value. "+1" is added because indexes start from 1, not 0.

In summary, this method generates a token for Turkish phrases and then takes the vocabulary size for the token and then prints the vocabulary size.

```
ottoman_tokenizer = tokenization(ottoman_turkish[:, 1])
ottoman_vocab_size = len(ottoman_tokenizer.word_index) + 1

ottoman_length = 8
print('Ottoman Vocabulary Size: %d' % ottoman_vocab_size)
```

```
Ottoman Vocabulary Size: 50487
```

The same operations were carried out for the Ottoman Turkish.

Note: This ss was taken while working on the entire dataset.

We converted the rows to numeric values. This allows the neural network to perform operations on the input data.

```
def encode_sequences(tokenizer, length, lines):
    seq = tokenizer.texts_to_sequences(lines)
    seq = pad_sequences(seq, maxlen=length, padding='post')
    return seq
```

This code defines a function called "encode_sequences" that takes in three parameters: a tokenizer, a length, and a list of lines.

The function first uses the "texts_to_sequences" method of the tokenizer to convert the input lines into numerical values, where each word is mapped to its corresponding integer value from the vocabulary index built earlier.

The next line calls the "pad_sequences" function from the Keras library on the output of the "texts_to_sequences" method. This function pads the sequences with 0 values so that all sequences have the same length, which is specified by the "length" parameter. The padding is done at the end of the sequence (post)

In summary, this code defines a function that takes in tokenizer, a length, and a list of lines, it then converts the lines into numerical values and pads the sequences with 0 values so that all sequences have the same length.

```
from sklearn.model_selection import train_test_split
train, test = train_test_split(ottoman_turkish, test_size=0.2, random_state = 12)
```

We divide our dataset into two separate parts as test and train. The test_size parameter is what percentage of the dataset is test.

```
# prepare training data
trainX = encode_sequences(ottoman_tokenizer, ottoman_length, train[:, 1])
trainY = encode_sequences(turkish_tokenizer, turkish_length, train[:, 0])

# prepare validation data
testX = encode_sequences(ottoman_tokenizer, ottoman_length, test[:, 1])
testY = encode_sequences(turkish_tokenizer, turkish_length, test[:, 0])
```

encode_sequences function is tokenizing the input sentences and encoding them into a numerical format that can be used as input for a neural network model. With these, test and train datasets are prepared.

```python
# build NMT model
def define_model(in_vocab,out_vocab, in_timesteps,out_timesteps,units):
    model = Sequential()
    model.add(Embedding(in_vocab, units, input_length=in_timesteps, mask_zero=True))
    model.add(LSTM(units))
    model.add(RepeatVector(out_timesteps))
    model.add(LSTM(units, return_sequences=True))
    model.add(Dense(out_vocab, activation='softmax'))
    return model
```

This function creates a neural machine translation (NMT) model using the Keras library. It takes in several parameters such as vocabulary size, number of units, and timesteps and creates the model by adding Embedding layer, LSTM layer, RepeatVector layer, Dense layer, and so on. The function returns the created NMT model.

```python
rms = optimizers.RMSprop(lr=0.001)
model.compile(optimizer=rms, loss='sparse_categorical_crossentropy')
```

The optimizer is set to RMSprop with a learning rate of 0.001. RMSprop is an optimization algorithm that helps to minimize the loss function during training.

```python
# train model
history = model.fit(trainX, trainY.reshape(trainY.shape[0], trainY.shape[1], 1),
                    epochs=100, batch_size=128, validation_split = 0.2,callbacks=[checkpoint],
                    verbose=1)
```

Epochs parameter is set to 100, meaning the model will go through the entire training dataset 100 times.

batch_size parameter is set to 128, meaning that 128 samples will be used in each update of the model's parameters.

validation_split parameter is set to 0.2, meaning that 20% of the training data will be used as validation data during training.

verbose parameter is set to 1, meaning that progress information will be printed to the console while the model is training.

```python
import numpy as np
model = load_model('model.h1.24_jan_19')
preds = model.predict(testX.reshape((testX.shape[0],testX.shape[1])))
pred_classes = np.argmax(preds, axis=1)
```

This code loads a pre-trained model and uses it to make predictions on the testX dataset. Then it selects the most likely class for each time step in the predicted output sequence.

```python
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['train','validation'])
plt.show()
```

This code is plotting the training and validation loss during the training process of the NMT model.

Validation loss is a measure of how well a model is able to generalize to new data.

A lower validation loss indicates that the model is better at generalizing to new data and is therefore less likely to overfit the training data. The result is obtained with test data.

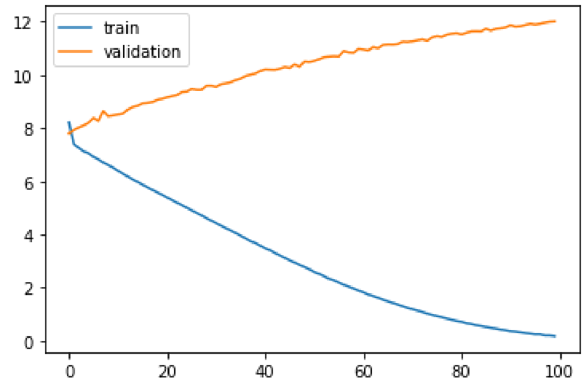
BLEU Score:

The BLEU score is calculated based on the number of n-grams (phrases of length 1 to n) in the machine-generated translation that match n-grams in the reference translations. The more n-grams that match, the higher the BLEU score. The score is usually between 0 and 1, with 1 indicating a perfect match between the machine-generated translation and the reference translation(s).

The BLEU Score was applied to the 30000 row dataset and the result in the 3rd test case was obtained.

TEST CASES :

1) Tried using 5000 rows dataset and 100 epochs

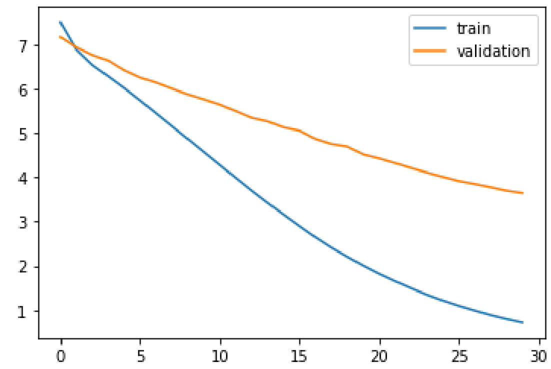


	actual	predicted
0	en doğru inancin bu olduğunu bu olacağini doğal olarak kabul etmek uygun olur	ve
1	bunun için bu bildirinin yayimlanip dağitilmasina aracilik etmeyi yararli bulmuyoruz	ve
2	gerçekten tam ornek ciftliği yakinlarinda karşimiza çikan bir otomobilin içinden tali bey göründü	ve
3	her zaman ölçülü davranilmali ve durumun iyi idare edilmesini yeniden salik veririm diyor	ve
4	vali paşa karargahimin sağlik işleri başkani olup evvelce örgüt için sivasa göndermiş olduğum tali beyi çağirip bu görevin yapilmasini ondan rica etmiş ve önlemler alinir alinmaz kendisinin de bi...	ve
5	hükumetin cemal paşa araciliğiyla yazdiklari bizim ileri sürdüğümüz düşünceler bir kez daha gözden geçirilmeye değer kanisindayim	ve
6	mandaterlik hakkinda ayrintili bilgiler posta ile gönderilmek üzeredir	ve
7	generalin sorduğu sorudan ana amacinin ne olabileceğini araştirmak istemedim	ve
8	ben bu umutla kendimi yüreklendiriyorum	ve
9	türlü nedenlerden ötürü suriyeden başka bu sözünü ettiğim illerimizi de işgal altinda bulunduran fransizlarin da bizimle anlaşmaya eğilimli olduklari anlaşilmakta idi	ve
10	işin din bilimi yönüne gelince hoca efendiler hiç kaygilanip üzülmesinler	ve

10	işin din bilimi yönüne gelince hoca efendiler hiç kaygilanip üzülmesinler	ve
11	her şey bitmiş ulusal amaç elde edilmiş değildir	ve
12	bu olaylar sirasinda ingilizler 25 haziran 1920 mudanyaya ve 2 temmuz 1920 de de bandirmaya birer askeri birlik çikardilar	ve
13	ordumuz 28 eylul sabahi ileri harekete geçti	ve
14	ferit beye durumu anlattiniz mi	
15	kendisi istanbul hükumeti tarafindan istanbula çağrildi	ve
16	ben nurettin paşa için uygulanmasi istenen işleme katilmadim	ve
17	sinop mebusu hakki hami beyin de hizla cezalandirilmalari isteğinde direnmesi bravo seslerile karşilaniyordu	ve
18	son günlerdeki bildirilerimizin gereğinin tamamile yapilip yapilmadiğini sordum	ve
19	celalettin arif ve hüseyin avni beylerin erzuruma varişlarindan sonra celalettin arif beyden 10 ve 15 16 ve 16 eylul 1920 tarihlerinde üç şifre telgraf aldim	ve

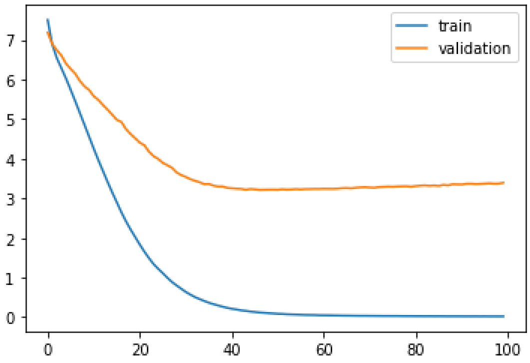
2) Tried using 30000 rows dataset and 30 epochs

```
Epoch 28/30
146/150 [============================>.] - ETA: 0s - loss: 0.8830
Epoch 28: val_loss improved from 3.84963 to 3.77549, saving model to model.h1.24_jan_19
WARNING:absl:Found untraced functions such as lstm_cell_8_layer_call_fn, lstm_cell_8_layer_call_and_return
150/150 [==============================] - 15s 98ms/step - loss: 0.8869 - val_loss: 3.7755
Epoch 29/30
146/150 [============================>.] - ETA: 0s - loss: 0.7963
Epoch 29: val_loss improved from 3.77549 to 3.69830, saving model to model.h1.24_jan_19
WARNING:absl:Found untraced functions such as lstm_cell_8_layer_call_fn, lstm_cell_8_layer_call_and_return
150/150 [==============================] - 15s 98ms/step - loss: 0.7997 - val_loss: 3.6983
Epoch 30/30
146/150 [============================>.] - ETA: 0s - loss: 0.7169
Epoch 30: val_loss improved from 3.69830 to 3.64653, saving model to model.h1.24_jan_19
WARNING:absl:Found untraced functions such as lstm_cell_8_layer_call_fn, lstm_cell_8_layer_call_and_return
150/150 [==============================] - 15s 100ms/step - loss: 0.7206 - val_loss: 3.6465
```



3) Tried with 30000 rows dataset and 100 epochs

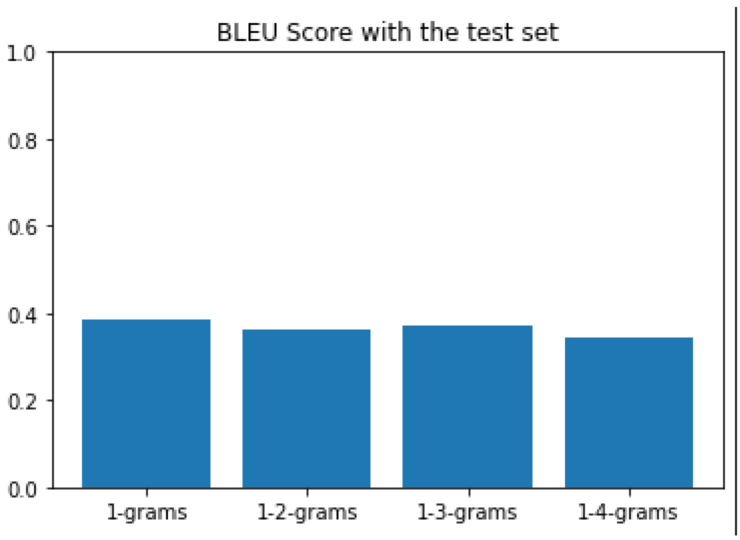
```
Epoch 98/100
146/150 [============================>.] - ETA: 0s - loss: 0.0223
Epoch 98: val_loss did not improve from 3.21632
150/150 [==============================] - 2s 13ms/step - loss: 0.0221 - val_loss: 3.3712
Epoch 99/100
146/150 [============================>.] - ETA: 0s - loss: 0.0217
Epoch 99: val_loss did not improve from 3.21632
150/150 [==============================] - 2s 13ms/step - loss: 0.0217 - val_loss: 3.3735
Epoch 100/100
146/150 [============================>.] - ETA: 0s - loss: 0.0220
Epoch 100: val_loss did not improve from 3.21632
150/150 [==============================] - 2s 13ms/step - loss: 0.0219 - val_loss: 3.3961
```



bu üç tür kararin gerekçeleri vermiş olduğum açiklamalar arasinda vardir	tür kararin gerekçeleri vermiş olduğum açiklamalar arasinda vardir
bu inanca aykiri görüş ve düşünceleri açiğa vuracaklarin vay haline	aykiri görüş ve düşünceleri açiğa vuracaklarin vay haline
görüyordum iki üç bin mil açiktan bakarak	üç ayni üç girmelerinin son damlanin
hayir kendin bak	hayir kendin bak
kuzum eşek nali yapsan bir usta çingenenin	kuzum eşek nali yapsan bir çingenenin
cetelerimizden bir kismi yok ediliyor	bir şeyi içinde istemiş

cünkü sorun gerçekten önemli ve ölüm kalim sorunu niteliğindeydi	cünkü bunlar ortadan güven elverişli olabilir
649	
ben bu ödevin yapilmasinin bir görev ve yetki sahibi olmaya bağli olduğunu ileri sürdüm	ve tam içinde olmasi zorunlu verdik
bana bu söylediğim şeyleri tümüyle kabullendi	bana bu söylediğim şeyleri tümüyle kabullendi
adirncla bir dikilir azminin gelir önüne	bir üyesi ancak yüz kan
efendiler mecliste yenilgiye uğrayanlarin gazeteci arkadaşlari bu sonucu elbette hiç beğenmediler	uğrayanlarin gazeteci arkadaşlari bu sonucu elbette hiç beğenmediler
kimin uğruna ölmüş desin de inleniesin	kimin uğruna ölmüş desin de inleniesin
kendilerine söylediğim sözleri olduğu gibi meclise söylemekte sakinca görmediğimi bildirdim	milli iyi cumhuriyet yapilmasi vermek güçleri olduğu söyledi

malatyada bulunan on ikinci atli alay komutanini da 78 eylul gecesi ben kendim telgraf başina çağirmiş ve görüşmekte idim	adapazari kaymakami bey tarafindan görevi ve sinirli anladim
canini feda edeceksin can düşmanin için	canini feda edeceksin can düşmanin için
rauf bey millet kaderini kendisinden başka bir kimseye birakmayi kendisi için alçaklik saydi dedikten sonra milletin ulusal egemenliği kayitsiz şartsiz olarak yürüten büyük millet meclisini k...	iyi ve doğru bir yönetim yöntemi olduğunu söylüyor
ikinci madde içeriğine gelince bu konu düşünülmeye değer ve türlü yönlerde tartişilmaya elverişlidir	konu düşünülmeye değer ve türlü yönlerde tartişilmaya elverişlidir
asil şaşilacak şey bundan sonra görüldü	orada güçlere böyle de böyle
diş durum istanbulda şöyle görünüyor fransa italya ingiltere türkiyede mandaterlik işini amerika senatosuna resmi olarak önermiş olmakla birlikte bütün güçlerini senatonun kabul etmemesi için ...	birlikte bütün güçlerini senatonun kabul etmemesi için harciyorlar



CONCLUSION:

Test cases have been tested with datasets of various sizes and different parameters. In small datasets, the validation value of the model does not decrease and increases gradually. This means that the model progresses by heart and supports this in the results it prints. When the amount of dataset grows, the validation level has progressed more meaningfully. The model is accurate and results close to the truth. When I tried to reduce the number of epochs, the model passed the training set less often and the validation value did not reach the saturation point.

Note:

Larger datasets can be run at higher capacity ram levels and better results can be obtained.

REFERENCES

https://www.kaggle.com/code/harshjain123/machine-translation-seq2seq-lstms/notebook

https://www.kaggle.com/code/databeru/machine-translation-fr-en-with-bleu-score