به نام خدا

مهدی فیروزبخت

۴۰۰۱۳۱۰۲۷

تمرین سری هشتم

درس شبکه های عصبی

کدهای این تمرین در فایل 400131027_HW08. Ipynb قرار داده شده است.

۱) در ابتدا کتابخانه های مورد نیاز را وارد کرده ایم. سپس با استفاده از کد زیر داده ها را فراخوانی کرده ایم. در این تمرین برای کاهش زمان انجام کدها از GPU و همچنین از ۱۰٪ از داده ها استفاده شده است :

```
data, metadata = tfds.load('ted_hrlr_translate/pt_to_en', with_info=True, as_supervised=True , split=['train[:10%]', 'test[:10%]', 'validation[:10%]'])
train , test, valid = data[0], data[1] , data[2]
```

سپس در مرحله بعد برای ساختن توکن از توکنایزر BERT که در tensorflow قرار داده شده است استفاده میکنیم :

```
model_name = 'ted_hrlr_translate_pt_en_converter'
tf.keras.utils.get_file(
    f'{model_name}.zip',
    f'https://storage.googleapis.com/download.tensorflow.org/models/{model_name}.zip',
    cache_dir='.', cache_subdir='', extract=True
)
```

```
'./ted_hrlr_translate_pt_en_converter.zip'
```

```
tokenizers = tf.saved_model.load(model_name)
```

سپس در مرحله بعد ، سایز Buffer ، Batch و MAX_TOKENS را مشخص میکنیم. حداکثر توکن به معنی این است که جمله هایی که بیشتر از مقدار حداکثر توکن داشته باشند را در نظر نمیگیرد. این حالت را با استفاده از تابع زیر پیاده سازی میکنیم :

```
def filter_max_tokens(pt, en):
  num_tokens = tf.maximum(tf.shape(pt)[1],tf.shape(en)[1])
  return num_tokens < MAX_TOKENS
```

در مرحله بعد نیاز داریم تا داده ها را به صورت دسته ای در اورده تا برای استفاده در مدل اماده باشد برای این کار از قطعه کد زیر استفاده میکنیم :

```
def tokenize_pairs(pt, en):
    pt = tokenizers.pt.tokenize(pt)
    pt = pt.to_tensor()
    en = tokenizers.en.tokenize(en)
    en = en.to_tensor()
    return pt, en
```

در مرحله آخر نیاز است تا داده های آموزش و اعتبارسنجی را برای استفاده آماده کنیم پس از قطعه کد زیر استفاده میکنیم :

```
def make_batches(ds):
  return (
      ds
      .cache()
      .shuffle(BUFFER_SIZE)
      .batch(BATCH_SIZE)
      .map(tokenize_pairs, num_parallel_calls=tf.data.AUTOTUNE)
      .filter(filter_max_tokens)
      .prefetch(tf.data.AUTOTUNE))


train_batches = make_batches(train)
val_batches = make_batches(valid)
```

۲)

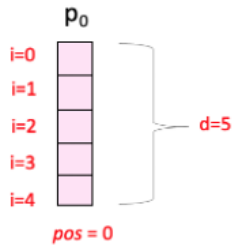
برای پیاده سازی تعبیه مکانی ابتدا نیاز داریم آن را تعریف کنیم :

در معماری ترانسفورماتور، از رمزگذاری موقعیتی برای دادن زمینه نظم به معماری غیر تکراری توجه چند سر استفاده می شود. بیایید این جمله را کمی باز کنیم. هنگامی که شبکه های تکراری با ورودی های توالی تغذیه می شوند، ترتیب ترتیبی (ترتیب گام های زمانی) به طور ضمنی توسط ورودی تعریف می شود. با این حال، لایه توجه چند سر در ترانسفورماتور یک لایه پیش‌خور است و یک دنباله کامل را به یکباره می‌خواند. همانطور که توجه بر روی هر نقطه داده (زمان-گام) به طور مستقل محاسبه می شود، زمینه ترتیب بین نقاط داده از بین می رود و توجه به ترتیب توالی تغییر نمی کند. همین امر به طور کلی برای سایر معماری‌های غیر تکراری مانند لایه‌های کانولوشنال که در آن‌ها فقط یک زمینه کوچک ترتیب متوالی وجود دارد، محدود به اندازه هسته کانولوشن، صادق است.

برای کاهش این مشکل، از مفهوم رمزگذاری موقعیتی استفاده می شود. این چیزی نیست جز افزودن یک تانسور (هم شکل توالی ورودی) با ویژگی های خاص به دنباله ورودی. تانسور رمزگذاری موقعیتی باید به گونه ای باشد که تفاوت مقدار مراحل خاص در دنباله با فاصله گام های جداگانه در زمان (توالی) مرتبط باشد. در فرمول ذکر شده ، pos موقعیت در زمان است، dmodel تعداد ابعاد و i شاخص بعد در تانسور ورودی است. به این ترتیب هر طول موج یک پیشرفت هندسی از ۲π تا k۲π است.

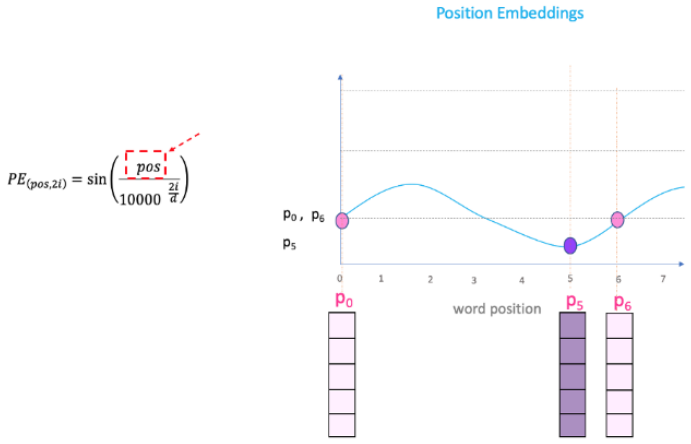
برای درک آن میتوانیم از مثال زیر استفاده کنیم :

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$



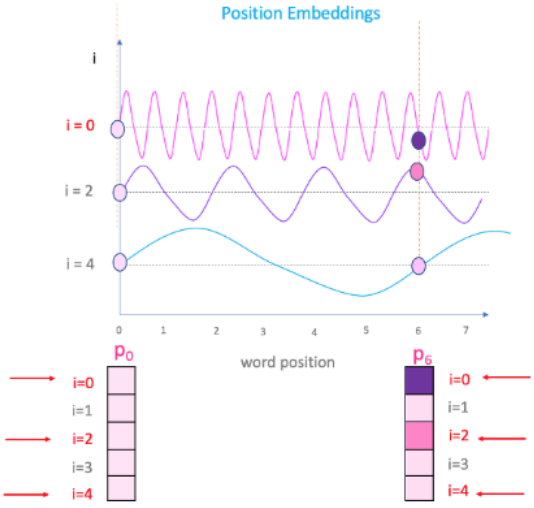
در اینجا "pos" به موقعیت "کلمه" در جمله اشاره دارد. P0 به جاسازی موقعیت کلمه اول اشاره دارد. "d" به معنای اندازه کلمه / نشانه است. در این مثال d=5. در نهایت، "i" به هر یک از ۵ بعد جداگانه تعبیه (یعنی۰ ، ۱ ، ۲ ، ۳ ، ۴) اشاره دارد. در حالی که "d" ثابت است، "pos" و "i" متفاوت است. بیایید سعی کنیم دو مورد بعدی را درک کنیم.

اگر منحنی sin را رسم کنیم و "pos" را (در محور x) تغییر دهیم، با مقادیر موقعیت متفاوت در محور y خواهید آمد. بنابراین، کلمات با موقعیت های مختلف، مقادیر جاسازی موقعیت متفاوتی خواهند داشت. هر چند مشکلی وجود دارد. از آنجایی که منحنی گناه در فواصل زمانی تکرار می شود، می توانید در شکل زیر ببینید که P0 و P6 با وجود اینکه در دو موقعیت بسیار متفاوت قرار دارند، مقادیر جاسازی موقعیت یکسانی دارند. اینجا جایی است که قسمت «i» در معادله وارد عمل می شود.



اگر "i" را در معادله بالا تغییر دهید، دسته ای از منحنی ها با فرکانس های متفاوت دریافت خواهید کرد. با خواندن مقادیر جاسازی موقعیت در برابر فرکانس های مختلف، مقادیر متفاوتی در ابعاد جاسازی مختلف برای P0 و P6 به دست می آید.

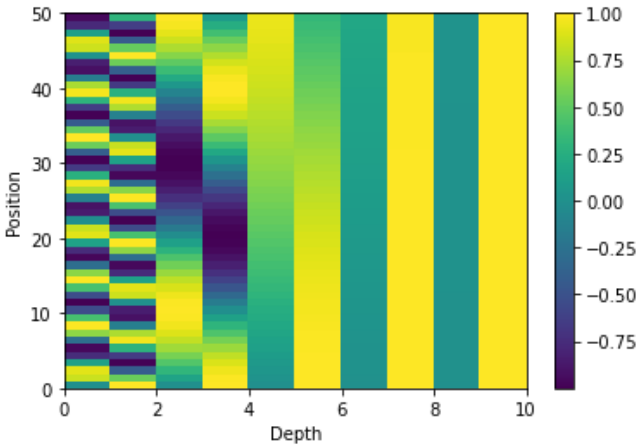
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$



برای روشن تر شدن این موضوع، در اینجا یک مثال خیالی نوشته شده است که این جدول تعبیه مکانی را برای یک تانسور ورودی خیالی با ۵۰ مرحله زمانی و ۱۲ کانال ترسیم می کند:

```
pos_encoding = positional_encoding(50, 10)
print (pos_encoding.shape)

plt.pcolormesh(pos_encoding[0])
plt.xlabel('Depth')
plt.xlim((0, 10))
plt.ylabel('Position')
plt.colorbar()
plt.show()
```

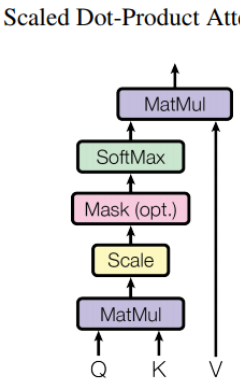
```
(1, 50, 10)
```



قطعه کد زیر ، پیاده سازی موقعیت مکانی را نشان میدهد :

```
def positional_encoding(position, d_model):
  angle_rads = (1 / np.power(10000, (2 * (np.arange(d_model)[np.newaxis, :]//2))
                / np.float32(d_model)) * np.arange(position)[:, np.newaxis])

  angle_rads[:, 0::2] = np.sin(angle_rads[:, 0::2])
  angle_rads[:, 1::2] = np.cos(angle_rads[:, 1::2])

  pos_encoding = angle_rads[np.newaxis, ...]

  return tf.cast(pos_encoding, dtype=tf.float32)
```

در ادامه به بررسی دقیق کد میپردازیم بتوانیم به سوالات مطرح شده پاسخ دهیم :



تابع توجه که توسط یک ترانسفورماتور استفاده می شود، سه ورودی می گیرد: Q (پرس و جو)، K (کلید)، V (مقدار). معادله ای که برای محاسبه وزن توجه استفاده می شود:

$$Attention(Q, K, V) = softmax_k\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

dot-product attention با ضریب ریشه دوم عمق مقیاس می شود. این کار به این دلیل انجام می‌شود که برای مقادیر زیاد عمق، محصول نقطه‌ای بزرگ‌تر می‌شود و تابع softmax را فشار می‌دهد، جایی که دارای گرادیان‌های کوچک است که منجر به یک softmax بسیار سخت می‌شود.

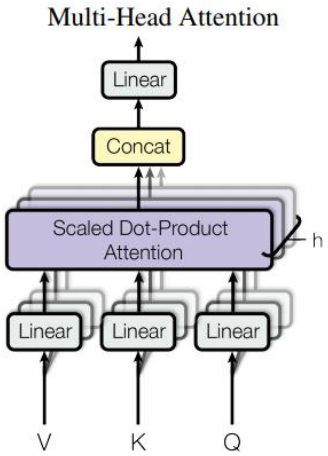
به عنوان مثال، در نظر بگیرید که Q و K دارای میانگین ۰ و واریانس ۱ هستند. ضرب ماتریس آنها میانگین ۰ و واریانس dk خواهد داشت. بنابراین ریشه دوم dk برای مقیاس بندی استفاده می شود،

بنابراین بدون توجه به مقدار dk، یک واریانس ثابت به دست می آورید. اگر واریانس خیلی کم باشد، خروجی ممکن است برای بهینه سازی موثر خیلی صاف باشد. اگر واریانس بیش از حد بالا باشد، softmax ممکن است در زمان اولیه اشباع شود و یادگیری را دشوار کند.

ماسک در -e91 ضرب می شود (نزدیک به بی نهایت منفی). این کار به این دلیل انجام می شود که ماسک با ضرب ماتریس مقیاس بندی شده Q و K جمع می شود و بلافاصله قبل از softmax اعمال می شود. هدف این است که این سلول ها را صفر کنیم و ورودی های منفی بزرگ به softmax در خروجی نزدیک به صفر هستند. از آنجایی که نرمال سازی softmax در امتداد ابعاد کلیدها انجام می شود، مقادیر توجه میزان اهمیت داده شده به کلیدها برای هر پرس و جو را تعیین می کند. خروجی نشان دهنده ضرب وزن توجه و بردار V (مقدار) است. این تضمین می‌کند که توکن‌هایی که می‌خواهید روی آنها تمرکز کنید همانطور که هست نگه داشته می‌شوند و توکن‌های نامربوط پاک می‌شوند.

در تابع scaled_dot_product_attention مقدار خروجی و وزن توجه محاسبه میشود.

در مرحله بعد کلاس مربوط به Multi-head attention را بررسی میکنیم :



توجه چند سر از چهار بخش تشکیل شده است:

Linear layers

Scaled dot-product attention

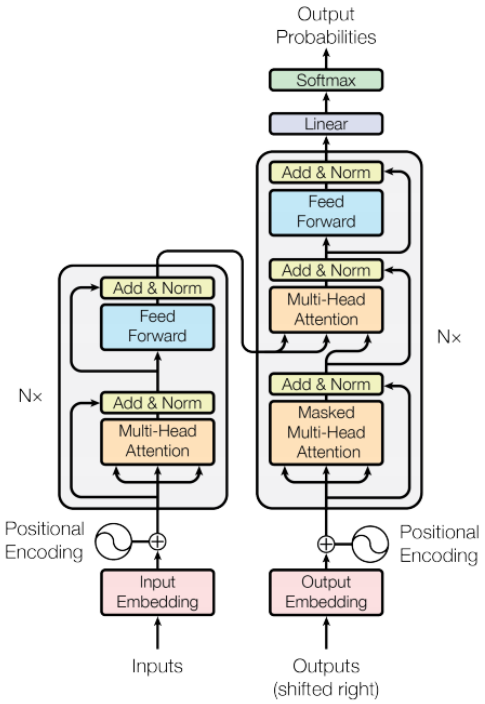
Final linear layer

هر بلوک توجه چند سر سه ورودی دریافت می کند. Q (پرس و جو)، K (کلید)، V (مقدار). اینها قبل از تابع توجه چند سر از طریق لایه های خطی (متراکم) قرار می گیرند.

تابع scaled_dot_product_attention که در بالا تعریف شده است، در یک تماس واحد اعمال می شود که برای کارایی پخش می شود. در مرحله توجه باید از ماسک مناسب استفاده شود. سپس خروجی توجه برای هر سر به هم متصل می شود (با استفاده از tf.transpose و tf.reshape) و یک لایه متراکم نهایی اعمال میگردد. به جای یک سر توجه واحد، Q، K و V به سرهای متعدد تقسیم می‌شوند، زیرا به مدل اجازه می‌دهد به طور مشترک به اطلاعات از زیرفضاهای نمایش متفاوت در موقعیت‌های مختلف توجه کند. پس از تقسیم، هر سر دارای ابعاد کاهش یافته است، بنابراین کل هزینه محاسباتی همان توجه یک سر با ابعاد کامل است.

برای شبکه جلورو از تابع زیر استفاده میکنیم :

```
def point_wise_feed_forward_network(d_model, dff):
  return tf.keras.Sequential([
      tf.keras.layers.Dense(dff, activation='relu'),
      tf.keras.layers.Dense(d_model)
  ])
```



برای بخش Encoder and decoder جمله ورودی از طریق N لایه رمزگذار منتقل می شود که یک خروجی برای هر نشانه در دنباله ایجاد می کند. رمزگشا به خروجی رمزگذار و ورودی خود (توجه به خود) برای پیش بینی کلمه بعدی توجه می کند.

ابتدا لایه رمزگذار :

هر لایه رمزگذار از زیر لایه ها تشکیل شده است:

1. Multi-head attention (with padding mask)
2. Point wise feed forward networks.

هر یک از این زیرلایه ها دارای یک اتصال باقی مانده در اطراف خود هستند و به دنبال آن یک لایه عادی سازی می شود. اتصالات باقیمانده به جلوگیری از مشکل ناپدید شدن گرادیان در شبکه های عمیق کمک می کند. خروجی هر زیرلایه ((LayerNorm(x + Sublayer(x است. نرمال سازی در محور d_model (آخرین) انجام می شود. در یک ترانسفورماتور N لایه رمزگذار وجود دارد.

کد های این بخش در کلاس EncoderLayer قرار داده شده است.

سپس لایه رمزگشا :

هر لایه رمزگشا از ۳ بخش تشکیل شده است :

1. Masked multi-head attention (with look ahead mask and padding mask)
2. Multi-head attention (with padding mask)
3. Point wise feed forward networks

هر یک از این زیرلایه ها دارای یک اتصال باقی مانده در اطراف خود هستند و به دنبال آن یک لایه عادی سازی می شود. خروجی هر زیرلایه ((LayerNorm(x + Sublayer(x است. نرمال سازی در محور d_model (آخرین) انجام می شود. تعدادی لایه رمزگشا در مدل وجود دارد. همانطور که Q خروجی را از اولین بلوک توجه رمزگشا دریافت می کند، و K خروجی رمزگذار را دریافت می کند، وزن توجه نشان دهنده اهمیت داده شده به ورودی رمزگشا بر اساس خروجی رمزگذار است. به عبارت دیگر، رمزگشا با نگاه کردن به خروجی رمزگذار و توجه به خروجی خود، توکن بعدی را پیش‌بینی می‌کند.

کد های این بخش در کلاس Decoder layer قرار داده شده است.

رمزگذار :

رمزگذار شامل موارد زیر است:

1. Input Embedding
2. Positional Encoding
3. N encoder layers

ورودی از طریق یک جاسازی قرار می گیرد که با رمزگذاری موقعیتی جمع می شود. خروجی این جمع، ورودی لایه های رمزگذار است. خروجی انکودر ورودی رسیور است.

کد این بخش در قسمت Encoder قرار دارد.

رمزگشا

رسیور شامل موارد زیر است:

1. Output Embedding
2. Positional Encoding
3. N decoder layers

هدف از طریق یک جاسازی قرار می گیرد که با رمزگذاری موقعیتی جمع می شود. خروجی این جمع، ورودی لایه های رمزگشا است. خروجی رسیور ورودی لایه خطی نهایی است.

کد این بخش در قسمت Decoder قرار دارد.

حال مدل اصلی را ایجاد میکنیم . یک ترانسفورماتور از رمزگذار، رمزگشا و یک لایه خطی نهایی تشکیل شده است. خروجی رسیور ورودی لایه خطی است و خروجی آن برگردانده می شود.

کد این بخش در قسمت کلاس Transformer قرار داده شده است.

برای افزایش سرعت این مدل ، مقادیر num_layers، d_model، dff کاهش یافته است. مدل پایه توصیف شده در مقاله استفاده شده است: num_layers=6، d_model=512، dff=2048 که در این کد از مقدار زیر استفاده شده است :

```
num_layers = 4
d_model = 128
dff = 512
num_heads = 8
dropout_rate = 0.1
```

برای قسمت بهینه سازی نیز از یک بهینه ساز Adam تغییریافته استفاده میشود :

```
optimizer = tf.keras.optimizers.Adam(learning_rate, beta_1=0.9, beta_2=0.98, epsilon=1e-9)
```

سپس در مرحله انتهایی با استفاده از قطعه کد زیر این مدل را آموزش میدهیم :

```
train_step_signature = [
    tf.TensorSpec(shape=(None, None), dtype=tf.int64),
    tf.TensorSpec(shape=(None, None), dtype=tf.int64),
]


@tf.function(input_signature=train_step_signature)
def train_step(inp, tar):
  tar_inp = tar[:, :-1]
  tar_real = tar[:, 1:]

  with tf.GradientTape() as tape:
    predictions, _ = transformer([inp, tar_inp],
                                 training = True)
    loss = loss_function(tar_real, predictions)

  gradients = tape.gradient(loss, transformer.trainable_variables)
  optimizer.apply_gradients(zip(gradients, transformer.trainable_variables))

  train_loss(loss)
  train_accuracy(accuracy_function(tar_real, predictions))
```

این مدل را با ۳۰ تکرار آموزش میدهیم و به دقت زیر میرسیم :

```
Loss 3.1790 Accuracy 0.4016  Validation Accuracy 0.2929
```

و بعد از ۵۰ تکرار به حالت زیر رسیده است که یک حالت بیش برازش است و از ۳۰ تکرار برای ادامه کارها استفاده میشود :

```
Loss 0.9987 Accuracy 0.7441  Validation Accuracy 0.2998
```

در مرحله آخر نیاز به یک مترجم داریم تا ورودی را دریافت کرده ، توکن آن را ساخته ، وارد مدل آموزش دیده کند و سپس نتیجه نهایی را وارد معکوس توکن کننده کرده و ترجمه آن را ارائه دهد. همچنین این مترجم باید وزن های Attention را به عنوان خروجی ارائه نماید تا برای استفاده در نقشه حرارت مورد استفاده قرار گیرد. این مدل توضیح داده شده در کلاس Translator نوشته شده است.

۳)

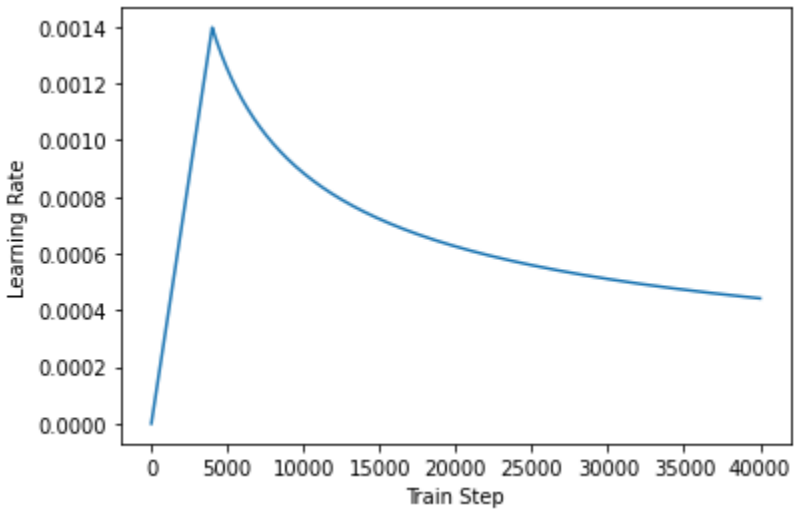
الف) طبق مقاله Attention Is All You Need که در مورد ترانسفورماتور و استفاده از attention است ، از یک بهینه ساز adam با beta1=0.9، beta2=0.98 ، epsilon=10e-9 استفاده شده است. همچنین طبق فرمول زیر که در این مقاله ذکر شده است لازم است learning rate در طول زمان تغییر کند :

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

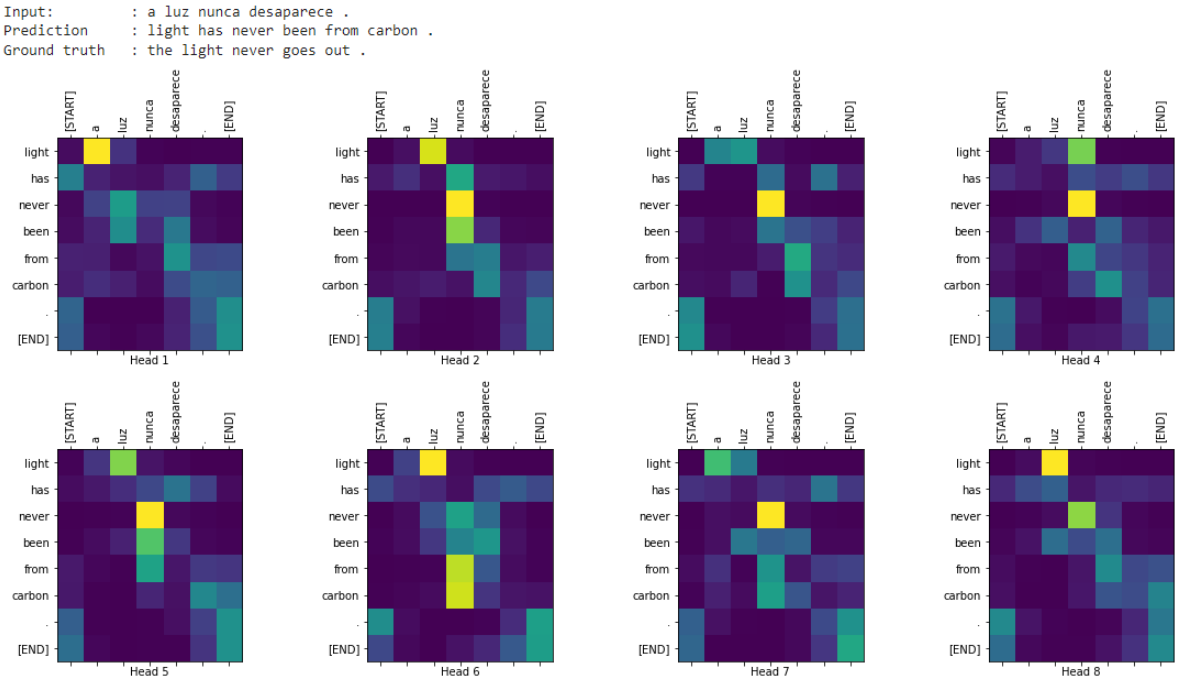
که در کد نویسی برای پیاده سازی این LR از تابع زیر استفاده شده است :

```
class CustomSchedule(tf.keras.optimizers.schedules.LearningRateSchedule):
  def __init__(self, d_model, warmup_steps=4000):
    super(CustomSchedule, self).__init__()

    self.d_model = d_model
    self.d_model = tf.cast(self.d_model, tf.float32)

    self.warmup_steps = warmup_steps

  def __call__(self, step):
    arg1 = tf.math.rsqrt(step)
    arg2 = step * (self.warmup_steps ** -1.5)

    return tf.math.rsqrt(self.d_model) * tf.math.minimum(arg1, arg2)
```

که پیاده سازی دقیق این فرمول است. در این فرمول با گذشت زمان و تا رسیدن تعداد مرحله های آموزش به تعداد warmup_steps ، LR رو به افزایش است و بعد از گذشت آن به سمت کاهش میرود. نمودار زیر نشان دهنده این بحث است :



ب)



برای جمله ورودی :

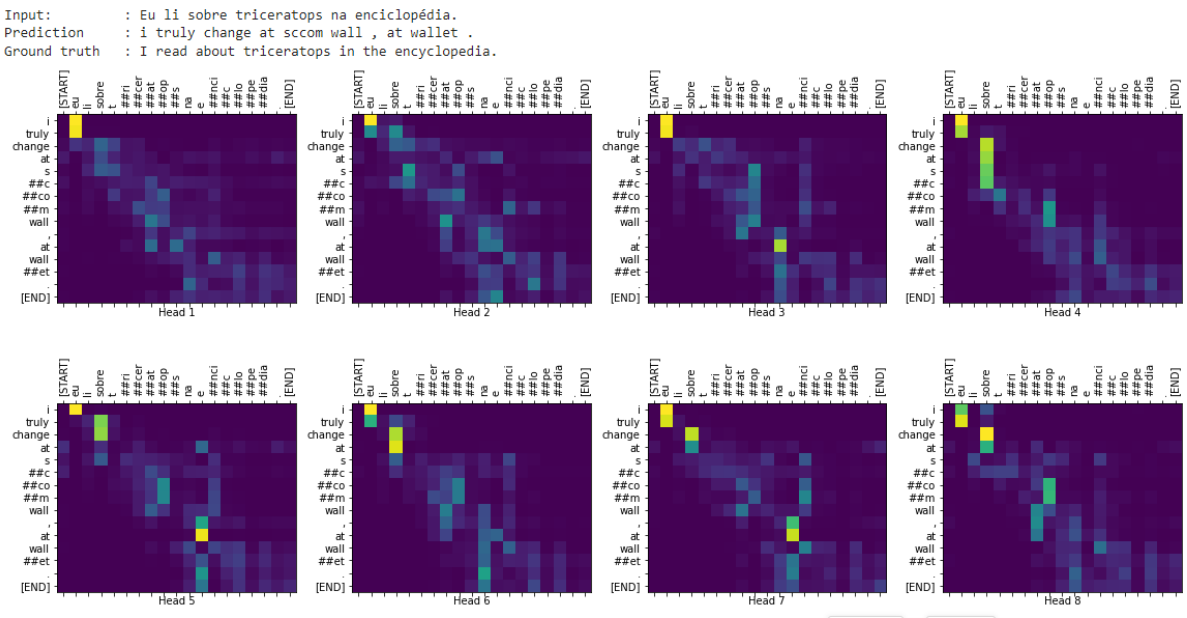
```
a luz nunca desaparece .
```

که معنای اصلی آن به صورت :

```
the light never goes out .
```

در بیشتر سر ها ، کلمه luz را به کلمه light ارتباط داده اند . برای کلمه nunca نیز در ۶ سر از ۸ سر به کلمه never ارتباط داده است. برای کلمه desaparece اما نتوانسته کلمه درستی را ارتباط دهد. همچنین قابله مشاهده است در حداقل ۴ سر از ۸ سر کلمه been را به کلمه nunca ارتباط داده است که این ارتباط اشتباه است. همچنین قابل مشاهده است که ۳ کلمه been , from , carbon در اکثر سر ها ارتباطی با کلمه desaparece و nunca برقرار کرده است و به نوعی این کلمات را به یکدیگر ارتباط داده است.

در کنار این جمله ، جمله دیگری را که جمله با کلمات جدیدی است را به این شبکه میدهیم و نتیجه زیر را ایجاد میکند.



در این مثال شبکه به خوبی سعی کرده تا کلمه جدید triceratops را ترجمه کند در حالی که در هنگام آموزش کلمه encyclopedia که معنی کلمه triceratops را ندیده است ، اما سعی کرده با ریز کردن کلمه ، کلمه را شناسایی کند. در این تمرین از تعداد کمتری کلمه نسبت با حالت اصلی استفاده شده است پس منطقی است که جملات ساخته شده ایرادات معنایی در آنها وجود داشته باشد زیرا در این تمرین بسیاری از کلمات در دایره لغات مدل دیده نشده است. اما مدل به خوبی سعی میکند حتی کلمات جدید را ترجمه کند.

ج)

1 text is :

Input: : depois , podem fazer-se e testar-se previsões .

Prediction : and then you can do this to do in violence .

Ground truth : then , predictions can be made and tested .

2 text is :

Input: : forçou a parar múltiplos laboratórios que ofereciam testes brca .

Prediction : they took home for cleaner , rainy - nose .

Ground truth : it had forced multiple labs that were offering brca testing to stop .

3 text is :

Input: : as formigas são um exemplo clássico ; as operárias trabalham para as rainhas e vice-versa .

Prediction : the scores does one of an area means you move to work with it and critical price .

Ground truth : ants are a classic example ; workers work for queens and queens work for workers .

4 text is :

Input: : uma em cada cem crianças no mundo nascem com uma doença cardíaca .

Prediction : a connectome of every single connectome with a connectome of a heart called your heart .

Ground truth : one of every hundred children born worldwide has some kind of heart disease .

5 text is :

Input: : neste momento da sua vida , ela está a sofrer de sida no seu expoente máximo e tinha pneumonia .

Prediction : in this moment , she ' s all his life , she looked at least site and she would have to get her mouth , she ' s like the pun quartzoost .

Ground truth : at this point in her life , she 's suffering with full-blown aids and had pneumonia .

6 text is :

Input: : onde estão as redes económicas ?

Prediction : where are these edph ?

Ground truth : where are economic networks ?

7 text is :

Input: : (aplausos)

Prediction : (applause)

Ground truth : (applause)

8 text is :

Input: : eu usei os contentores de transporte , e também os alunos ajudaram-nos a fazer toda a mobília dos edifícios , para torná-los confortáveis , dentro do orçamento do governo , mas também com a mesma a área da casa , mas muito mais confortável .

Prediction : i realizeds of the personal again , and the other thing to take people in those children with all of them , but musicians and musicians and musicians and even better right code , even better code , but the same time has translated into the same time , they ' re doing the same time , they ' re doing these days .

Ground truth : i used the shipping container and also the students helped us to make all the building furniture to make them comfortable , within the budget of the government but also the area of the house is exactly the same , but much more comfortable .

9 text is :

Input: : e , no entanto , a ironia é que a única maneira de podermos fazer qualquer coisa nova é dar um passo nessa direção .

Prediction : and yet , i finally has to look at this as an important thing to do that you know that kind of trying to get an adulthood .

Ground truth : and yet , the irony is , the only way we can ever do anything new is to step into that space .

10 text is :

Input: : a luz nunca desaparece .

Prediction : light has never been from carbon .

Ground truth : the light never goes out .

11 text is :

Input: : `` agora , `` " tweets " " , quem está a `` " tweetar " " ? "

Prediction : ` ` now , ` ` ' ' you will stro ? ' ' ' ' this : a forest ! ' '

Ground truth : now , tweets , who 's tweeting ?

12 text is :

Input: : no egito , por exemplo , 91 % das mulheres que vivem hoje no egito foram mutiladas sexualmente dessa forma .

Prediction : on india , it ' s applicable that china that are consumers actually , in the materials ; they have been able to teach that they have the absolute numbers .

Ground truth : in egypt , for instance , 91 percent of all the females that live in egypt today have been sexually mutilated in that way .

13 text is :

Input: : por outro lado , os bebés de 15 meses ficavam a olhar para ela durante muito tempo caso ela agisse como se preferisse os brócolos , como se não percebessem a situação .

Prediction : on the other day , babies might be asked anyone around with her , having long as she will feel her understanding as a patient .

Ground truth : on the other hand , 15 month-olds would stare at her for a long time if she acted as if she liked the broccoli , like they could n't figure this out .

14 text is :

Input: : naquele momento , percebi quanta energia negativa é precisa para conservar aquele ódio dentro de nós .

Prediction : that moment , i realized that , very badness is been thinking about the whole food chain .

Ground truth : in that instant , i realized how much negative energy it takes to hold that hatred inside of you .

15 text is :

Input: : e a discussão é : o que é que isso interessa .

Prediction : it ' s an original thing that is that it does that matter .

Ground truth : and the discussion is , who cares ? right ?

16 text is :

Input: : se escolhermos um lugar e formos cuidadosos , as coisas estarão sempre lá quando as procurarmos .

Prediction : if you go to a gps thing , you could become partial things , and you would surely find the experience of them .

Ground truth : if you designate a spot and you 're scrupulous about it , your things will always be there when you look for them .

17 text is :

Input: : é um museu muito popular agora , e criei um grande monumento para o governo .

Prediction : it ' s a very interesting place in this way , and i looked at home .

Ground truth : it 's a very popular museum now , and i created a big monument for the government .

18 text is :

Input: : é completamente irrelevante .

Prediction : and it ' s rather rather rather d rather .

Ground truth : it 's completely irrelevant .

19 text is :

Input: : todos defenderam que a sua técnica era a melhor , mas nenhum deles tinha a certeza disso e admitiram-no .

Prediction : all of them were given to us to be realized that they had their community , but they were not fully control the rules .

Ground truth : `` they all argued that , `` " my technique is the best , " " but none of them actually knew , and they admitted that . "

20 text is :

Input: : a partir daquele momento , comecei a pensar .

Prediction : so i of that there ' s where i think of this .

Ground truth : at that moment , i started thinking .