تمرین سری چهارم
درس یادگیری عمیق

در این تمرین هدف پیادسازی تسک پردازش گفتار با یک مدل end to end شبکه عصبی می‌باشد. مجموعه دادگان مورد استفاده در لینک زیر قرار دارند جهت آشنایی به لینک زیر مراجعه کنید.

voxforge dataset

یک فایل برای دانلود و پیش‌پردازش داده‌ها آپلود شده است.
کتابخانه مورد نیاز این تمرین pytorch می‌باشد. که ورژن های مورد نیاز در فایل توضیحات جداگانه آمده است.
توصیه: اجرای تسک‌های مربوط به speech نیازمند زمان آموزش و حافظه قابل توجهی هستند. بنابراین توصیه می‌شود از google colab استفاده شود.

مرحله‌ی 0

دانلود و پیش پردازش داده ها:

با استفاده از script موجود در فایل‌های دانلودی با اجرای کد download_prepare_data.sh داده های شما دانلود و پیش پردازش خواهند شد. برای جزییات بیشتر به فایل readme.md مراجعه کنید.

مرحله‌ی 1

معرفی ساختار شبکه

همانطور که می‌دانیم شبکه‌های موسوم به transformer networks با حذف لایه‌های بازگشتی و استفاده از لایه‌های feedforward و یک مدل attention جدید بسیار مورد توجه قرار گرفت.
در این تمرین می‌خواهیم تسک پردازش گفتار را با استفاده از یک شبکه transformer به صورت end to end پیاده سازی کنیم(برای سادگی قسمت decoder مربوط به شبکه transformer را در نظر نمیگیریم) .
برای پیادسازی این قسمت دقت داشته باشید که ابتدا دانلود و پیش‌پرازش را روی داده‌ها، توسط کد داده شده انجام می‌دهیم. سپس برای استفاده از مدل transformer داده های صوتی ابتدا وارد قسمت input embedding و سپس وارد Encoderمی‌شوند. و داده های متنی متناسب با داده صوتی داده شده به قسمت output embedding بدون وارد شدن به قسمت decoder وارد تابع هزینه ctc میشود).
ساختار شبکه transformer به صورت زیر می‌باشد: (برای درک بهتر و جزییات بیشتر این مقاله مطالعه شود.)

Transformer Network Modules:

- Input Embedding
- Output Embedding (پیادسازی لازم نیست)
- Positional Encoding
- Encoder
- Decoder(پیادسازی لازم نیست)

برای embed کردن داده‌های ورودی صوتی از ساختار شبکه زیر استفاده می‌کنیم.

Input Embedding (emb_cnn feature embedding):

(0): Conv2d(1,32, kernel_size=(41, 11), stride=(2, 2), padding=(0, 10))
(1): BatchNorm2d(32, eps=1e-05, momentum=0.1)

(2): Hardtanh(min_val=0, max_val=20)
(3): Conv2d(32,32, kernel_size=(21, 11), stride=(2, 1))
(4): BatchNorm2d(32, eps=1e-05, momentum=0.1)
(5): Hardtanh(min_val=0, max_val=20)

Positional Encoding:

برای این پیادهسازی این قسمت به بخش Positional Encoding مربوط به این مقاله مراجعه کنید.

Encoder:

(1): Dropout(p=0.1, inplace=False)
(2): Linear(in_features=dim_input, out_features=dim-model)
(3): LayerNorm((dim-model,), eps=1e-05)
(4): PositionalEncoding()
(5) (num-layers-encoder) * Encoder Layer

هر Encoder Layer به صورت زیر میباشد.

Encoder Layer:

(1) Self Attention
(2) POS_FFN

ساختارهای POS_FFN, Self Attention در زیر آمده است :

Self Attention:

MultiHeadAttention(
(1): (query_linear): Linear(in_features=dim-model, out_features=num_heads*dim_key)
(2): (key_linear): Linear(in_features=dim-model, out_features=num_heads * dim_key)
(3): (value_linear): Linear(in_features=dim-model,out_features=num_heads*dim_value)
(4): (Attention): ScaledDotProductAttention()
(5): LayerNorm((dim-model,), eps=1e-05,)
(6): Linear(in_features=num_heads* dim_value, out_features=dim-model)
(7): (dropout): Dropout(p=0.1))

POS_FFN:

PositionwiseFeedForwardWithConv(
(1): Conv1d(dim-model, dim-inner, kernel_size=(1,), stride=(1,))
(2): Conv1d(dim-inner, dim-model, kernel_size=(1,), stride=(1,))
(3): Dropout(p=0.1)
(4): LayerNorm((dim-model,), eps=1e-05))

Attention:

ScaledDotProductAttention(
(1): (dropout): Dropout(p=0.1)
(2): (softmax): Softmax(dim=2))

تابع هزینه (loss function):

برای بهینه سازی مدل از تابع هزینه CTC loss استفاده کنید. (برای torch با ورژن ۱.۱.۰ به بعد این تابع پیاده سازی شده است و نیازی به پیادهسازی مجدد آن نیست اما دانستن نحوه کارکرد همهی قسمتها مهم میباشد)لینک زیر توضیح مختصری در این باره است
towardsdatascience

تابع بهینه‌ساز (optimizer): برای بهینه‌سازی از تابع بهینه‌ساز Adam با پارامتر های زیر استفاده کنید

$$adam\ optimizer: \ \beta_1 = 0.9,\ \beta_2 = 0.98,\ \varepsilon = 10^{-9}$$

و طبق فرمول زیر learning rate را در طول زمان آموزش تغییر دهید:

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

که step_num شماره مرحله آموزش است. Warmup_steps =4000

مرحله‌ی 2

ساختار کلی شبکه به این صورت می‌باشد که برای سادگی کار در شبکه transformer قسمت decoder را در نظر نمی‌گیریم. فرض کنیم داده‌های ورودی (صوتی) ابتدا با کد داده شده تبدیل به spectrogram میشوند و بعد از ان وارد یک شبکه برای embedding میشوند و پس از ترکیب با قسمت positional encoding وارد یک شبکه encoder با ساختار توضیح داده شده در بالا می‌شود و با عبور از لایه encoder به همراه خروجی متنی وارد تابع هزینه ctc می‌شود و مقدار هزینه محاسبه می‌شود.

الف:

سه معیار WER و CER و Loss را در طول آموزش محاسبه و در یک نمودار برای داده های آموزشی(train) و داده‌های اعتبارسنجی (validation) رسم کنید.

ب:

برای چند داده ورودی (۱۰ داده ورودی صوتی)دنباله خروجی متنی حاصل از دنباله ورودی تولید شده توسط مدل را در هر 50 گام محاسبه و نمایش دهید.

برای دیکد کردن دنباله از روش best path decoding استفاده کنید.

پارامترهای مدل را به صورت زیر در نظر بگیرید.

```
Sample_rate = 16000,
Window_size = 0.02
hidden_size = int(math.floor( (sample_rate * window_size) / 2) + 1)
hidden_size = int(math.floor(hidden_size - 41) / 2 + 1)
hidden_size = int(math.floor(hidden_size - 21) / 2 + 1)
dim_input = hidden_size * 32
num_heads= 8
dim_model =256
dim_key=64
dim-emb= 256
```

dim_value= 64
dim-inner= 1024
num-layers_decoder = 4
Learning rate= 1e-4
batch-size= 12

نکات کلی تمرین:

- در صورت مشاهده مشابهت کد بین هر دو دانشجو، نمره تمرین هر دو نفر صفر لحاظ خواهد شد.
- در صورت مشاهده هرگونه مشابهت کد با کدهای موجود در اینترنت، نمره تمرین صفر لحاظ خواهد شد. اگر بخشی از کد که از قسمت‌های اصلی تمرین نمی‌باشد را کدهای آماده استفاده می کنید حتما لینک و منبع آن را اعلام کنید.
- لازم به ذکر است نیمی از نمره تمرین مربوط به گزارش می‌باشد. بنابراین رعایت اصول نگارشی حائز اهمیت می‌باشد.
- در نوشتن گزارش، لحاظ جزئیات گزارش الزامی است مانند موارد زیر
  - ارجاع دادن به مطالب و اشکالی که از مقاله و وبسایت ها گرفته شده اند
  - توضیح اشکال و جداول در caption
  - نوشتن فرمول و قرار ندادن عکس‌های فرمول در متن
  - نوشتن نتایج شبیه‌سازی ها به صورت جدول و شکل
  - درست بودن متن از لحاظ قواعد دستور زبانی و نگارشی
  - موارد تکمیلی در فایل template آمده است.
- گزارش تمرین را به صورت فایل pdf و در کنار کدهای تمرین در سایت آپلود نمایید.
- نحوه نامگذاری به صورت studentnumber_Homeworknumber.pdf باشد.
- برای هرگونه پرسش پیرامون این تمرین را با ایمیل‌های aliparchekan@gmail.com, esmaeilfarhang@gmail.com مطرح نمایید.