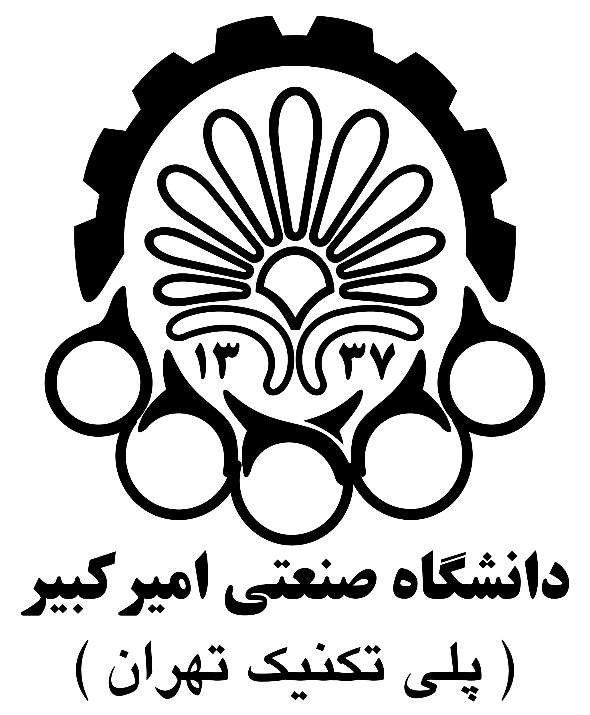
****

**دانشکده مهندسی کامپیوتر**

**تمر ین دوم درس فهم زبان**

**دکتر زینلی**

**غلامرضا دار 400131018**

**بهار 1401**

فهرست مطالب

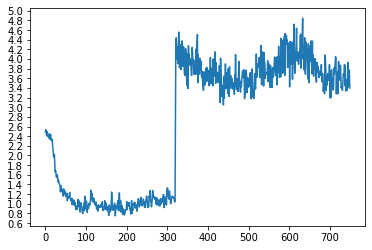
[1) انکدر دیکدر با اتنشن 3](#_Toc118473056)

[2) ترنسفورمر 6](#_Toc118473057)

# 1) انکدر دیکدر با اتنشن

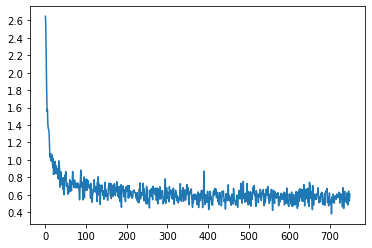
برای پیاده­سازی تبدیل­کننده Grapheme2Phoneme با استفاده از معماری انکدر دیکدر با اتنشن، از نمونه کد ترجمه ماشینی زبان انگلیسی به فرانسوی موجود در [وبسایت PyTorch](https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html) استفاده کردیم.

پس از انجام تغییرات لازم جهت آماده سازی مدل برای خواندن داده فارسی داده شده در صورت سوال، مدل را آموزش دادیم. نمودار لاس مدل اولیه را در تصویر زیر مشاهده میکنید.



نمودار Loss مدل اولیه انکدر دیکدر

پس از مقداری آزمایش و تحقیق به این نتیجه رسیدیم که دلیل این اتفاق، استفاده از بهینه ساز SGD بود. این بهینه ساز به تغییرات ناگهانی گرادیان بسیار حساس است و در این مثال نیز احتمالا با دیدن یک گرادیان خاص از یک بهینه محلی خارج شده و چند مرحله بعد یک بهینه محلی دیگر را یافته است. با تغییر بهینه­ساز به Adam و همچنین کوچک کردن نرخ یادگیری، مدل به درستی آموزش یافت.



نمودار Loss، پس از تغییر Optimizer

**راهنمایی آموزش مدل:**

درون پوشه encoder\_decoder\_attention یک فایل به نام train.py وجود دارد. پس از اطمینان از نصب بودن jiwer و pytorch و دسترسی به cuda، با استفاده از دستور زیر آموزش مدل را شروع کنید.

$ python train.py

**راهنمایی تست مدل:**

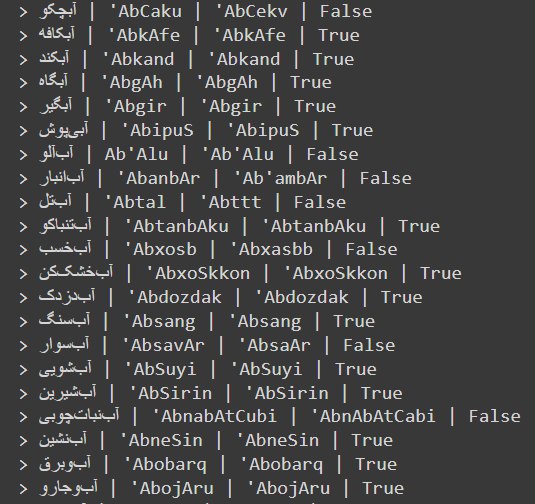
درون پوشه encoder\_decoder\_attention یک فایل به نام test.py وجود دارد. پس از اطمینان از نصب بودن jiwer و pytorch و دسترسی به cuda، با استفاده از دستور زیرارزیابی مدل را شروع کنید. با این کار، از وزن­های از قبل ذخیره شده برای مدل استفاده خواهید کرد.

$ python test.py

نتیجه نهایی بدست آمده به صورت زیر است:

|  |  |
| --- | --- |
| Word Error Rate | Phoneme Error Rate |
| 65% | 17% |

با توجه به خروجی های مدل به این نتیجه می­رسیم که این مدل، در اکثر کلمات، تعداد 1 یا 2 حرف را اشتباه پیش بینی می­کند و این باعث می­شود WER بسیار بالا باشد.



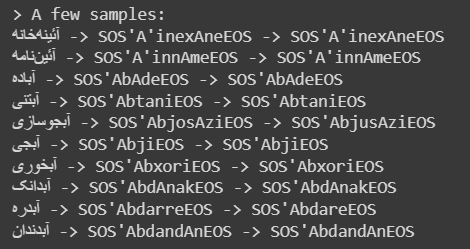
نمونه هایی از خروجی مدل انکدر دیکدر با اتنشن

# 2) ترنسفورمر

برای پیاده سازی این بخش از سوال از نمونه پیاده سازی مشابه برای یک مجموعه داده خارجی استفاده کردیم. پس از انجام تغییرات لازم جهت آماده سازی مدل برای کار با مجموعه داده­ی فارسی سوال، به نتایج بهتری نسبت به مدل انکدر دیکدر با اتنشن رسیدیم.

نتیجه نهایی به­دست آمده به کمک معماری ترنسفورمر را در جدول زیر مشاهده می­کنید.

|  |  |
| --- | --- |
| Word Error Rate | Phoneme Error Rate |
| 27% | 4.93% |



تصویر نمونه هایی از خروجی این مدل

**راهنمایی آموزش مدل:**

درون پوشه transformer یک فایل به نام train.py وجود دارد. پس از اطمینان از نصب بودن python-Levenshtein و pytorch و دسترسی به cuda، با استفاده از دستور زیر آموزش مدل را شروع کنید.

$ python train.py

**راهنمایی تست مدل:**

درون پوشه transformer یک فایل به نام test.py وجود دارد. پس از اطمینان از نصب بودن python-Levenshtein و pytorch و دسترسی به cuda، با استفاده از دستور زیرارزیابی مدل را شروع کنید. با این کار، از وزن­های از قبل ذخیره شده برای مدل استفاده خواهید کرد.

$ python test.py