به نام خدا

دانشکده سامانه‌های هوشمند

روش‌های یادگیری ماشین در پردازش زبان طبیعی (۸۳۰۴۳۶۸)

نیم‌سال اول ۱۴۰۵-۱۴۰۴

تمرین شماره۵

تاریخ تحویل:
۱۴۰۴/۱۰/۱۸

---

۱. (۵٪) **[پژوهش – مفاهیم شبکه‌های عصبی بازگشتی]** در درس با مفاهیم مختلف شبکه‌های بازگشتی آشنا شدید. در این بخش می‌خواهیم برخی مفاهیم دیگر در این حوزه را بررسی کنیم.

أ. پیش‌بینی ساختاریافته[۱] توضیح دهید و فرق آن با طبقه‌بندی[۲] بررسی کنید این تکنیک در چه کاربردهایی استفاده می‌شود؟

ب. هم‌ترازی دنباله‌ای[۳] چیست و چگونه می‌توان از RNN برای این کار استفاده کرد؟ کاربردهای آن را در حوزه‌هایی مانند بیوانفورماتیک[۴] بررسی کنید.

ت. مدل‌های ترکیبی که شبکه‌های بازگشتی را با معماری‌های دیگر مانند CNN یا Transformer ترکیب می‌کنند، چه مزایا و کاربردهایی دارند؟ توضیح دهید هرکدام از شبکه‌های عصبی چه نقشی در دسته‌بندی می‌توانند داشته باشند.

ث. روش تدریس اجباری[۵] را در شبکه‌های عصبی بازگشتی توضیح دهید.

---

۲. (۱۰٪) **[نظری – مدل‌های مبدل، شبکه‌های بازگشتی، شبکه‌های پیچشی]**

سه مدل دنباله‌ای با طول دنباله n و اندازه نهان d در نظر بگیرید،

- یک Vanilla RNN/LSTM
- یک 1 D CNN عمیق شامل L لایه، هر کدام با اندازه هسته k ، گام ۱ و بدون dilation ؛
- یک لایه رمزگذار مبدل با self-attention کامل (بدون ماسک) به‌همراه یک شبکه پیشروِ موضعی[6]

أ. برای هر معماری، حداقل تعداد عملیات متوالی (گام‌های زمانی یا لایه‌ها) لازم را که اطلاعات در جایگاه i بتواند بر جایگاه j اثر بگذارد، یعنی طول مسیر برهم‌کنش $P(i,j)$ ، تعیین کنید.برای CNN ، مقدار P(i,j) را بر حسب $|i-j|$ ، k و L بنویسید و شرطی روی L و k بیان کنید که تحت آن هر دو جایگاه دلخواه در دنباله بتوانند با یکدیگر برهم‌کنش داشته باشند.به‌طور خلاصه توضیح دهید این طول‌های مسیر چگونه بر مدل‌سازی وابستگی‌های دوربرد و پدیده ناپدید شدن / انفجار گرادیان اثر می‌گذارند.

ب. پیچیدگی زمانی هر معماری را برای یک پاس پیشرو کامل از میان همه L لایه، در نماد Big-O و به‌عنوان تابعی از n، d و k به‌دست آورید و آنها را با هم مقایسه کنید.

ت. چرا مبدل‌ها در عمل برای دنباله‌های بلند روی GPU اغلب سریع‌تر از RNN ها هستند؟ توضیح دهید.

---

۳. (۵٪) **[نظری – بردار زمینه و توجه]** این پرسش ایده‌های مطرح‌شده در مدل‌های Seq2Seq و سازوکار توجه مدل‌های Bahdanau و Luong را بررسی می‌کند.

أ. در معماری Seq2Seq کلاسیکِ بدون توجه، رمزگذار[7] کل جمله ورودی را به یک بردار زمینه با طول ثابت تبدیل می‌کند. توضیح دهید چرا این بردار ثابت از دیدگاه انتقال اطلاعات یک گلوگاه اطلاعاتی محسوب می‌شود.

---

[۱] Structured prediction
[۲] Classification
[۳] Sequence Alignment
[۴] Bioinformatics
[۵] Teacher forcing
[6] position-wise feed-forward network.
[7] Encoder

h.veisi@ut.ac.ir

به نام خدا

ب. توضیح دهید این گلوگاه در عمل چگونه باعث افت عملکرد می‌شود، به‌ویژه وقتی طول جمله زیاد می‌شود.یک مثال کوتاه (مثلاً یک جمله بلند در ترجمه) بزنید و بگویید چه نوع اطلاعاتی ممکن است در انتهای دنباله در بازسازی/ترجمه دچار خطا شود.

ت. سازوکار «توجه» در Seq2Seq (به سبک Bahdanau یاLuong) را به‌صورت مفهومی توضیح دهید؛ در یک گام زمانی از رمزگشا [۱]امتیازدهی/هم‌ترازی چگونه نسبت به حالت‌های پنهان رمزگذار انجام می‌شود و وزن‌ها چگونه نرمال می‌شوند؟ همچنین بگویید بردار زمینه وابسته به زمان چگونه ساخته می‌شود.

ث. توضیح دهید این سازوکار دقیقاً چگونه به رمزگشا اجازه می‌دهد به‌جای تکیه بر یک بردار ثابت، در هر گام به تمام حالت‌های پنهان رمزگذار رجوع کند و عملاً «نگاه به عقب» داشته باشد.

---

٤. (۵٪) **[نظری – خود توجه]** سازوکار توجه (QKV attention) را می‌توان به‌صورت عملیاتی بر یک بردار پرسش $q \in R^d$، یک مجموعه از بردارهای مقدار $\{v_1, \dots, v_n\}$ با $v_i \in R^d$، و یک مجموعه از بردارهای کلید $\{k_1, \dots, k_n\}$ با $k_i \in R^d$ در نظر گرفت که به‌صورت زیر تعریف می‌شود:

$$c = \sum_{i=1}^{n} \alpha_i \, v_i, \qquad \alpha_i = \frac{\exp\left(k_i^{\top} q\right)}{\sum_{j=1}^{n} \exp\left(k_j^{\top} q\right)}$$

در این‌جا ضرایب $\alpha_1, \dots, \alpha_n$ را «وزن‌های توجه» می‌نامیم. توجه کنید که خروجی $c \in R^d$ یک میانگین وزن‌دار از بردارهای مقدار است که وزن‌ها توسط بردار α تعیین می‌شوند.

أ. سعی کنید مجموعه‌ای از بردارهای پرسش $\{q_1, \dots, q_n\}$ به‌همراه بردارهای مقدار و کلیدها طوری بسازید که خروجی c برای هر بردار پرسش ورودی q_i همیشه برابر با یکی از بردارهای مقدار v_j باشد، یعنی:

$$\forall\, q_i \implies \exists\, j \in \{1, \dots, n\} \text{ s.t. } c = v_j$$

توضیح دهید چه ویژگی‌هایی برای مجموعه بردارهای مقدار، بردارهای کلید و بردارهای پرسش مطلوب است تا این خواسته تحقق یابد.

ب. حالا حالت فقط دو بردار کلید و مقدار را در نظر بگیرید، یعنی $n = 2$ چه بردار پرسشی باعث می‌شود میزان توجه به هر دو بردار مقدار دقیقاً یکسان باشد یعنی $\alpha_1 = \alpha_2 = 0.5$

---

۵. (۵٪) **[پژوهش – مفاهیم مدل‌های زبانی بزرگ [۲]]** در درس با مفاهیم مدل‌های زبانی بزرگ آشنا شدید. در این بخش می‌خواهیم برخی مفاهیم دیگر در این حوزه را بررسی کنیم.

أ. درباره‌ی مدل‌های زبانی صوتی [۳] تحقیق کنید، نحوه کارکرد آن‌ها را توضیح دهید و چند نمونه از آن‌ها را معرفی کنید.

ب. مفهوم Mixture-of-Experts را توضیح دهید و نحوه‌ی کار آن‌ها را بررسی کنید.

ت. مدل‌های زبانی بزرگ مانند GPT و BERT را از نظر معماری، کاربردها، و محدودیت‌ها مقایسه کنید

ث. چرا استفاده از چندین سرِ توجه ۴ کوچک‌تر، نسبت به یک سازوکار توجهِ واحد با همان تعداد کل پارامترها، ممکن است مزیت داشته باشد.

---

[۱] Decoder

[2] Large Language Models (LLM)

[۳] Audio Language Models

[٤] Attention Head

h.veisi@ut.ac.ir

ج. سرهای مختلف چه نوع روابطی را می‌توانند میان توکن‌ها یاد بگیرند؟

---

۶. (۳۵٪) [پیاده‌سازی –ترجمه ماشین عصبی با شبکه‌های مبدل] در این پرسش، شما از شبکه‌های عصبی مبدل برای پیاده‌سازی یک سیستم ترجمه ماشین عصبی[۱] استفاده خواهید کرد. هدف شما این است که متنی را از یک زبان به زبان دیگر ترجمه کنید. برای این کار از مدل‌های مبدل و مجموعه داده‌ی این لینک استفاده خواهید کرد. در این تمرین بهتر است به‌جای یادگیری تصادفی بردارهای تعبیه[۲]، از بردارهای زبانی از پیش آموزش‌دیده (مانند FastText با بُعد ۳۰۰) برای هر دو زبان ورودی و خروجی استفاده کنید و لایه‌های تعبیه مدل را با این بردارها مقداردهی اولیه کنید.

أ. پیش‌پردازش داده‌ها

۱. تمیز کردن داده‌ها: وظایفی شامل حذف نویزها، کلمات توقف، نشانه‌ها، و کاراکترهای غیرضروری را بر مجموعه داده اعمال کنید.
۲. توکن‌سازی: متن را به توکن‌ها (کلمات یا جملات) و تبدیل آن‌ها به فرمت‌های قابل استفاده برای مدل تقسیم کنید.
۳. ایجاد دیکشنری‌ها: یک دیکشنری از کلمات و توکن‌ها برای مدل‌های ورودی و خروجی بسازید
۴. تقسیم داده‌ها: داده‌ها را به سه دسته آموزش، آزمون، و ارزیابی تقسیم کنید.
۵. Padding: ازآنجایی که طول همه جملات یکسان نیستند، Padding مناسبی برای ورودی شبکه در نظر بگیرید.این Padding می‌تواند در سمت راست جمله و یا در سمت چپ آن باشد.

ب. آموزش مدل

۱. مدل مبدل: معماری شبکه‌ی نهایی می‌تواند مانند جدول زیر باشد اما در صورتی که معماری بهتری مد نظر دارید از آن نیز می‌توانید استفاده کنید:

Output Shape	Layer Type
(None,)	Input (encoder_input)
(None, sequence_len, embed_dim)	TokenAndPositionEmbedding (encoder)
(None, sequence_len, embed_dim)	TransformerEncoder
(None, None, embed_dim)	Input (encoded_seq_input)
(None,)	Input (decoder_input)
(None, sequence_len, embed_dim)	TokenAndPositionEmbedding (decoder)
(None, sequence_len, embed_dim)	TransformerDecoder

---

[۱] Neural Machine Translation (NMT)
[۲] Embedding

h.veisi@ut.ac.ir

(None, sequence_len, embed_dim)	Dropout
(None, sequence_len, fr_vocab_size)	Dense (decoder_output)
(None, sequence_len, fr_vocab_size)	Output

جدول ۱ معماری شبکه

ت. ارزیابی مدل

۱. معیارهای ارزیابی:

- دقت ترجمه: میزان دقت ترجمه‌های مدل با استفاده از معیارهای همچون BLEU و ROUGE ارزیابی کنید.
- نمودار دقت و خطا در طول آموزش را رسم کنید.

۲. ارزیابی با روش‌های مبتنی بر مدل

- برای ارزیابی کیفیت ترجمه‌های خود، روی ۱۰۰۰ جمله اول مجموعه‌داده تست امتیاز BERTScore را با استفاده از مدل HooshvareLab/bert-base-parsbert-uncased محاسبه کنید. میانگین BERTScore روی این ۱۰۰۰ جمله را گزارش کنید.

---

۷. (۳۵٪) [پیاده‌سازی – تشخیص احساسات با شبکه‌های عصبی بازگشتی] در این تمرین، شما با استفاده از چهار روش دسته‌بندی شامل شبکه عصبی MLP، RNN، GRU و LSTM، تشخیص احساسات را تحلیل می‌کنید و نتایج این مدل‌ها را با یکدیگر مقایسه می‌کنید.

۱. پیش‌پردازش داده‌ها

أ. تمیز کردن متن: حذف نویزها، کلمات توقف[۱]، نشانه‌ها و کاراکترهای غیرضروری.

ب. توکن‌سازی[۲]: تقسیم متن به کلمات و توکن‌های مجزا.

ت. داده‌ها به سه دسته آموزش، آزمون و ارزیابی تقسیم کنید.

۲. استخراج ویژگی‌ها با روش‌های مختلف:

- Word2Vec:

بردارهای از پیش آموزش‌داده‌شده را از این لینک دریافت کنید.برای هر سند، میانگین و جمع بردارهای کلمات آن را محاسبه کرده و به‌عنوان نمایش برداری سند استفاده کنید.

- TF-IDF:

با استفاده از کتابخانه‌های موجود، ماتریس TF-IDF را برای اسناد ایجاد کنید.

- FastText:

[۱] Stop words
[۲] Tokenization

h.veisi@ut.ac.ir

به نام خدا

دانشکده سامانه‌های هوشمند

روش‌های یادگیری ماشین در پردازش زبان طبیعی (۸۳۰۴۳۶۸)

نیم‌سال اول ۱۴۰۵-۱۴۰۴

تمرین شماره۵

تاریخ تحویل:
۱۴۰۴/۱۰/۱۸

مدل FastText را بر روی دادگان آموزش داده و بردارهای کلمات را استخراج کنید. مانند Word2Vec، میانگین و جمع بردارهای کلمات هر سند را محاسبه کنید

۳. آموزش مدل

أ. شبکه‌ی عصبی ساده

a. از TF-IDF برای استخراج ویژگی‌ها استفاده کنید.

b. یک لایه Dense با تعداد مناسب نرون‌ها برای یادگیری روابط ترتیبی در داده‌ها

a. لایه Dense با یک نورون و تابع فعال‌سازی Sigmoid برای پیش‌بینی کلاس‌ها.

ب. مدل RNN: یک مدل RNN با معماری زیر طراحی کنید:

a. لایه Embedding برای تبدیل کلمات به بردارهای عددی.

b. یک لایه SimpleRNN با تعداد مناسب نرون‌ها برای یادگیری روابط ترتیبی در داده‌ها.

c. لایه Dense با یک نورون و تابع فعال‌سازی Sigmoid برای پیش‌بینی کلاس‌ها.

ت. مدل LSTM: یک مدل LSTM با معماری زیر طراحی کنید:

a. لایه Embedding برای تبدیل کلمات به بردارهای عددی.

b. یک لایه LSTM برای یادگیری روابط طولانی‌مدت بین کلمات.

c. لایه Dense با یک نورون و تابع فعال‌سازی Sigmoid برای پیش‌بینی کلاس‌ها.

ث. مدل GRU: یک مدل GRU با معماری زیر طراحی کنید:

a. لایه Embedding برای تبدیل کلمات به بردارهای عددی.

b. یک لایه GRU برای یادگیری روابط بین کلمات.

c. لایه Dense با یک نورون و تابع فعال‌سازی Sigmoid برای پیش‌بینی کلاس‌ها.

۴. ارزیابی مدل

أ. معیارهای ارزیابی

- چهار مدل را با استفاده از معیارهای زیر ارزیابی کنید:
  - دقت
  - AUC
  - F-1 socre

ب. مقایسه مدل‌ها

- عملکرد سه مدل را با استفاده از جدول یا نمودار مقایسه کنید.
- تحلیل کنید که کدام مدل عملکرد بهتری داشته و چرا.

ت. نتیجه‌گیری

- مشخص کنید که کدام روش برای این مسئله مناسب‌تر است.
- پیشنهادهایی برای بهبود مدل‌ها (مانند استفاده از داده‌های بیشتر یا مدل‌های پیشرفته‌تر) ارائه دهید.

h.veisi@ut.ac.ir

به نام خدا

دانشکده سامانه‌های هوشمند

روش‌های یادگیری ماشین در پردازش زبان طبیعی (۸۳۰۴۳۶۸)

نیم‌سال اول ۱۴۰۵-۱۴۰۴

تمرین شماره۵

تاریخ تحویل:
۱۴۰۴/۱۰/۱۸

---

۸. (۲۰٪ نمره اضافی) [پیاده‌سازی – تنظیم دقیق مدل‌های زبانی بزرگ] در این تمرین، هدف تنظیم دقیق یک مدل زبانی بزرگ برای وظیفه‌ی خلاصه‌سازی است. برای این منظور، از مدل Flan-T5 استفاده می‌کنیم. البته می‌توانید از سایر مدل‌های زبانی که سیستم شما قابلیت پردازش آن‌ها را دارد نیز بهره ببرید. همچنین مجموعه‌داده مورد استفاده از دستوری زیر قابلیت دسترسی دارد. از مجموعه‌داده‌ی زیر که حاوی اخبار و خلاصه‌ی آن‌ها می‌باشد برای این وظیفه استفاده می‌کنیم.

```
data = load_dataset("pn_summary")
```

أ. ارزیابی مدل بدون تنظیم دقیق

ابتدا مدل را بدون تنظیم دقیق برای این کار به دو روش زیر ارزیابی کنید:

a) خلاصه را با استفاده از Promptی که خودتان نوشتید در دو حالت زیر تولید کنید:
   - تولید خلاصه بدون دادن مثال به مدل
   - تولید خلاصه با دادن چند مثال به مدل

b) Prompt engineering: خلاصه را با استفاده از Promptهایی که برای مدل پیشنهاد می‌شوند (در اینجا تعدادی را می‌توانید مشاهده کنید) در دو حالت زیر تولید کنید:
   - تولید خلاصه بدون دادن مثال به مدل
   - تولید خلاصه با دادن چند مثال به مدل

ب. آموزش مدل

a) تنظیم دقیق معمولی
   - با استفاده از کلاس Trainer در کتابخانه‌ی Hugging Face، مدل را به روش استاندارد تنظیم دقیق کنید.
   - خروجی‌های مدل تنظیم‌شده را با خلاصه‌های انسانی و همچنین خلاصه‌های مدل اولیه (Base Model) مقایسه کنید.
   - تعداد پارامترهایی که در اینجا تنظیم دقیق می‌کنید را گزارش کنید.

b) تنظیم دقیق با استفاده از LORA
   - از روش LORA با کمک کتابخانه‌ی PEFT استفاده کنید.
   - خروجی‌های به‌دست‌آمده را با خلاصه‌های انسانی، مدل پایه، و مدل تنظیم‌شده‌ی بخش قبلی مقایسه کنید.
   - تعداد پارامترهایی که تنظیم دقیق می‌شوند را گزارش کنید.
   - رنک‌های مختلف برای LORA را آزمایش کنید و نتایج را مقایسه کنید.

در هر بخش، مقادیر معیارهای ROUGE و BLEU را گزارش کنید و مقایسه‌ای از ترجمه‌های مختلف انجام دهید.

h.veisi@ut.ac.ir