پاسخ سوالات مقاله خوانی

دانشکده مهندسی کامپیوتر

تکلیف چهارم درس مبانی NLP

استاد درس: دکتر برادران

دستیاران درس:

آئین کوپایی، هاجر مظاهری

سیاوش امیرحاجلو

993613007

تیر 1403

سوال اول) معماری wav2vec 2.0 و فرآیند پیش آموزش را با جزئیات توضیح دهید.

معماری wav2vec 2.0

1. رمزگذار ویژگی کانولوشنال (Convolutional Feature Encoder)

- هدف: شکل موج‌های صوتی خام را به نمایش‌های گفتاری نهفته نقشه می دهد.

- فرآیند: ورودی صوتی خام X توسط یک سری لایه‌های کانولوشن پردازش می شود که منجر به دنباله ای از نمایش‌های پنهان $z1,z2,\ldots,zT$ می شود. هر zt یک بخش از صدا را نشان می دهد.

2. ماژول کوانتیزاسیون (Quantization Module)

- هدف: بازنمایی‌های نهفته را گسسته می کند تا مجموعه ای از واحدهای گفتاری کوانتیزه ایجاد کند.

- مکانیسم: از کمی سازی محصول با چندین codebooks استفاده می کند. هر codebooks دارای چندین ورودی است و نمایش‌های نهفته با استفاده از Gumbel softmax به یکی از این ورودی‌ها نگاشت می شوند که انتخاب را متفاوت می کند.

- خروجی: دنباله ای از نمایش‌های نهفته کوانتیزه شده $q1,q2,\ldots,qT$.

3. شبکه ترانسفورماتور

- هدف: وابستگی‌های دوربرد در سیگنال صوتی را ضبط می کند.

- ساختار: متشکل از لایه‌های متعدد ترانسفورماتور با مکانیسم‌های خود توجه (self-attention).

- ورودی: نمایش‌های پنهان $z1,z2,\ldots,zT$ از رمزگذار ویژگی.

- خروجی: نمایش‌های متنی c1,c2,…,cT که برای کارهای پایین دستی استفاده می شود.

فرآیند پیش آموزش

1. نقاب زدن (Masking)

- فرآیند: به طور تصادفی بخشی (6.5٪) از خروجی‌های رمزگذار ویژگی پنهان را انتخاب می کند و آنها را ماسک می کند. هر بخش ماسک دار شامل 10 مرحله زمانی متوالی است.

2. کار متضاد (Contrastive task)

- هدف: نمایش نهفته کوانتیزه صحیح qt را برای یک مرحله زمانی پوشانده از مجموعه ای از حواس پرت کننده‌ها پیش بینی میکند.

- تابع هدف: ضرر متضاد (Contrastive loss)، به صورت زیر تعریف می شود:

$$L = -\log \frac{\exp(\text{sim}(c_t, q_t))}{\sum_{q' \in Q_t} \exp(\text{sim}(c_t, q'))}$$

که در آن sim(a,b) نشان دهنده شباهت کسینوس بین a و b است و Qt مجموعه ای از حواس پرتی (distractors) برای مرحله زمانی t است. .

3. مجازات تنوع کتاب کد (Codebook Diversity Penalty)

- هدف: استفاده از تمام ورودی‌های Codebook را برای اطمینان از نمایش متنوع تشویق می کند.

- مکانیسم: آنتروپی توزیع softmax متوسط را روی ورودی‌های کتاب کد در یک دسته از گفته‌ها به حداکثر می رساند:

$$\frac{1}{GV}\sum_{g=1}^{G}\sum_{v=1}^{V} p_{g,v}\log p_{g,v}$$

که در آن G تعداد کدها، V تعداد ورودی‌های هر Codebook و pg,v احتمال انتخاب ورودی v در Codebook g است.

4. دسته بندی چند زبانه

- فرآیند: نمونه‌های گفتار از چندین زبان برای تشکیل دسته‌های آموزشی استفاده می شود.

- Sampling: از توزیع چندجمله ای برای زبان‌های نمونه استفاده می کند و زبان‌های با منبع بالا و کم منبع را متعادل می کند.

5. جزئیات آموزش

- مدل‌ها: دو معماری "پایه" و "بزرگ" ارزیابی می شوند که در تعداد بلوک‌های ترانسفورماتور و ابعاد متفاوت هستند.

- پیاده سازی: در fairseq، با تنظیمات خاص برای برش و دسته بندی نمونه‌های صوتی پیاده سازی شده است.

سوال دوم) تفاوت مدل wav2vec2.0 و wav2vec xlsr-53 در چیست؟

مدل‌های wav2vec 2.0 و wav2vec XLSR-53 هر دو چارچوب‌های یادگیری خود نظارتی برای نمایش گفتار هستند که توسط Facebook AI توسعه یافته‌اند، اما اساساً در روش‌های آموزشی و کاربردهایشان متفاوت هستند.

wav2vec 2.0

1. هدف:

- برای یادگیری کلی خود نظارتی بازنمودهای گفتاری طراحی شده است.

2. معماری:

- رمزگذار ویژگی کانولوشن برای تبدیل شکل موج‌های صوتی خام به نمایش‌های گفتاری پنهان.

- ماژول Quantization برای گسسته سازی نمایش‌های نهفته.

- شبکه ترانسفورماتور برای گرفتن وابستگی‌های دوربرد و زمینه سازی بازنمایی‌های نهفته.

3. داده‌های آموزشی:

- از قبل روی مقدار زیادی از داده‌های گفتاری بدون برچسب (مانند Librispeech) آموزش دیده است.

4. مراحل قبل از آموزش:

- پوشاندن بخش‌های تصادفی نمایش‌های نهفته (latent representations) و حل یک کار متضاد (contrastive task) برای پیش بینی نمایش‌های کوانتیزه صحیح از مجموعه ای از حواس پرت کننده‌ها (distractors).

- از جریمه تنوع کتاب کد (codebook diversity penalty) برای تشویق استفاده از تمام ورودی‌های codebook استفاده می کند.

- پیش آموزش شامل یک زبان واحد یا مجموعه ای همگن از داده‌های گفتاری است.

wav2vec XLSR-53

1. هدف:

- طراحی شده برای یادگیری خود نظارتی بین زبانی، و آن را برای کارهای تشخیص گفتار چند زبانه مناسب می کند.

2. معماری:

- بر اساس معماری wav2vec 2.0 با اجزای مشابه ساخته شده است: رمزگذار ویژگی کانولوشن، ماژول کوانتیزاسیون و شبکه ترانسفورماتور.

3. داده‌های آموزشی:

- از قبل در 53 زبان مختلف آموزش دیده است، از این رو نام XLSR-53 (بازنمایی‌های گفتاری متقابل زبانی با 53 زبان) نامیده می شود.

- مجموعه داده متنوع تر است و زبان‌های مختلف با ویژگی‌های آوایی متفاوت را در بر می گیرد.

4. مراحل قبل از آموزش:

- ماسک کردن و رویکرد کار متضاد مشابه wav2vec 2.0.

- شامل دسته بندی چند زبانه، که در آن دسته‌ها از نمونه‌های گفتاری چندین زبان تشکیل می شوند.

- از توزیع چند جمله ای برای زبان‌های نمونه استفاده می کند و زبان‌های با منبع بالا و کم منبع را متعادل می کند تا تعمیم بین زبان‌ها را بهبود بخشد.

تفاوت‌های کلیدی

1. حوزه داده‌های آموزشی:

- wav2vec 2.0: آموزش داده شده در گفتار تک زبانه یا همگن.

- wav2vec XLSR-53: آموزش داده‌های چندزبانه شامل 53 زبان، که به طور خاص برای کارهای بین زبانی طراحی شده است.

2. قابلیت چند زبانه:

- wav2vec 2.0: در درجه اول برای یک زبان یا زبان‌های نزدیک به هم بهینه شده است.

- wav2vec XLSR-53: بهینه شده برای تشخیص و کار با انواع مختلف زبان‌ها، بهبود عملکرد در محیط‌های چند زبانه.

3. مورد استفاده:

- wav2vec 2.0: مناسب برای یادگیری بازنمایی گفتار خود نظارت عمومی، معمولاً در سناریوهایی استفاده می شود که داده‌های با کیفیت بالا برای یک زبان خاص در دسترس است.

- wav2vec XLSR-53: ایده آل برای برنامه‌های چند زبانه، مانند تشخیص گفتار بین زبانی و درک وظایف که در آن داده‌های آموزشی چندین زبان را در بر می گیرند.

سوال سوم) رمزگشایی در مدل wav2vec2.0 با چه الگوریتمی انجام میشود؟ روش را توضیح دهید.

مدل wav2vec 2.0 از یک الگوریتم رمزگشایی مبتنی بر ترکیبی از رمزگشای طبقه‌بندی زمانی ارتباطی (CTC) و گاهی اوقات برای بهبود عملکرد، یک مدل زبان (LM) برای تبدیل خروجی مدل به رونوشت‌های متنی استفاده می‌کند.

رمزگشایی با wav2vec 2.0

1. طبقه بندی زمانی اتصال (CTC)

رمزگشا CTC:

- مدل wav2vec 2.0 دنباله ای از احتمالات را روی مجموعه ای از نشانه‌ها (کاراکترها یا زیرکلمه‌ها) برای هر فریم از صدای ورودی خروجی می دهد.

- از CTC برای تراز کردن این احتمالات با رونویسی هدف استفاده می شود، که امکان توالی‌های ورودی با طول متغیر را فراهم می کند و مشکل داشتن فریم‌های بیشتر از توکن‌های خروجی را حل می کند.

- الگوریتم CTC یک نشانه "blank " ویژه را معرفی می کند که نمی تواند هیچ خروجی را برای یک فریم نشان دهد و به مدل اجازه می دهد از فریم‌های خاصی رد شود و تکرار نشانه خروجی را مدیریت کند.

- تابع ضرر CTC در طول تمرین برای بهینه سازی هم ترازی بین توالی پیش بینی شده و واقعی استفاده می شود.

2. رمزگشایی جستجوی پرتو (Beam Search Decoding)

جستجوی پرتو:

- جستجوی پرتو یک الگوریتم رمزگشایی پیشرفته است که رونویسی‌های چندگانه (فرضیه‌ها) ممکن را در هر مرحله پیگیری می‌کند و توالی‌های top-k را بر اساس احتمال تجمعی آنها حفظ می‌کند.

- این روش بین بررسی چند فرضیه و محدود کردن به محتمل ترین آنها تعادل برقرار می کند.

- عرض تیر (k) تعیین می کند که در هر مرحله چند فرضیه حفظ می شود. عرض پرتو بزرگتر به طور کلی دقت را به قیمت افزایش منابع محاسباتی بهبود می بخشد.

3. ادغام با یک مدل زبان (اختیاری اما رایج)

مدل زبانی (LM):

- یک مدل زبان خارجی (به عنوان مثال، یک مدل n-gram یا یک مدل زبان عصبی مانند GPT) می تواند در فرآیند رمزگشایی ادغام شود تا دقت رونویسی را با ارائه پیش بینی‌های آگاه از زمینه بهبود بخشد.

- نمرات LM با نمرات مدل آکوستیک (از wav2vec 2.0) برای رتبه‌بندی مجدد فرضیه‌های جستجوی پرتو، معمولاً از طریق جمع وزنی ترکیب می‌شوند.

فیوژن کم عمق:

- متداول ترین روش ادغام یک مدل زبان در فرآیند رمزگشایی، همجوشی سطحی است که در آن احتمالات لاگ LM به احتمالات لاگ مدل آکوستیک اضافه می شود که توسط پارامتر وزن کنترل می شود.

مراحل رمزگشایی با جزئیات

1. خروجی مدل

مدل wav2vec 2.0 شکل موج صوتی ورودی را پردازش می کند و دنباله ای از لجیت‌ها را که احتمال هر توکن (از جمله توکن خالی) را در هر مرحله زمانی نشان می دهد، خروجی می دهد.

2. رمزگشایی CTC

- logitها از یک تابع softmax عبور داده می شوند تا توزیع‌های احتمال به دست آید.

- الگوریتم CTC این احتمالات را با در نظر گرفتن همه ترازهای ممکن توالی ورودی به نشانه‌های خروجی رمزگشایی می کند و احتمالات همه ترازهای معتبر را جمع می کند.

- در طول استنتاج، رمزگشایی حریصانه CTC می تواند برای رمزگشایی سریع اما کمتر دقیق استفاده شود، جایی که محتمل ترین نشانه در هر مرحله زمانی به طور مستقیم انتخاب می شود.

3. جستجوی پرتو با CTC

- یک الگوریتم جستجوی پرتو برای کاوش چندین توکن نشانه (پرتوها) به طور همزمان اعمال می شود.

- در هر مرحله زمانی، الگوریتم دنباله‌های top-k را بر اساس نمرات ترکیبی مدل آکوستیک و زبان آنها پیگیری می کند.

4. ترکیب یک مدل زبان:

- در صورت استفاده از یک مدل زبان، احتمالات لاگ آن به امتیازات جستجوی پرتو اضافه می شود.

- وزن LM را می توان تنظیم کرد تا تأثیر مدل آکوستیک و مدل زبان را متعادل کند و کیفیت رونویسی نهایی را بهبود بخشد.

سوال چهارم) برای بهبود نتایج بدست آمده چه روش یا تکنیکی را پیشنهاد میکنید.

1. Fine-Tuning با داده‌های برچسب دار

- Fine-Tuning

از داده‌های گفتاری برچسب دار برای تنظیم دقیق مدل wav2vec 2.0 از پیش آموزش دیده استفاده کنید. این فرآیند پارامترهای مدل را تنظیم می‌کند تا با ویژگی‌های خاص مجموعه داده هدف هماهنگی بهتری داشته باشد و عملکرد را در وظایف یا زبان‌های خاص بهبود بخشد.

- داده‌های خاص دامنه

اگر برنامه هدف دارای ویژگی‌های خاصی باشد (مانند رونویسی پزشکی، هوش مصنوعی مکالمه)، مدل را روی داده‌های دامنه خاص تنظیم کنید.

2. افزایش داده‌ها (Augmentation)

- افزایش نویز

انواع مختلف نویز (نویز پس زمینه، نویز سفید) را به داده‌های آموزشی اضافه کنید تا مدل در سناریوهای دنیای واقعی قوی تر شود.

- اختلال سرعت

سرعت صدا را بدون تغییر زیر و بم آن تغییر دهید. این به مدل کمک می کند تا تغییرات در سرعت صحبت کردن را بهتر تعمیم دهد.

- SpecAugment

تکنیک‌های تقویت را مستقیماً روی طیف‌نگارها اعمال کنید، مانند زمان تابش، پوشش فرکانس و پوشش زمانی.

3. ادغام مدل زبان

- از یک مدل زبان قوی تر استفاده کنید

یک مدل زبان قدرتمندتر (به عنوان مثال، LM مبتنی بر ترانسفورماتور) را با رمزگشای wav2vec 2.0 ادغام کنید.

- LM متنی

استفاده از LMهای آگاه از زمینه که می توانند از اطلاعات اضافی، مانند قسمت قبلی مکالمه، برای بهبود دقت رونویسی استفاده کنند.

- تنظیم وزن LM

Fine-Tuning وزن‌های مدل زبان را در طول ادغام کم عمق تا تعادل بهینه بین نمرات مدل آکوستیک و مدل زبان را پیدا شود.

4. بهینه سازی جستجوی پرتو

- افزایش عرض پرتو

آزمایش با عرض پرتو بزرگتر در رمزگشایی جستجوی پرتو تا فرضیه‌های بیشتری را کشف شود و دقت بهبود پیدا کند.

- راهبردهای هرس

اجرای استراتژی‌های هرس برای مدیریت کارآمد بار محاسباتی با حفظ دقت بالا.

5. آموزش انتقالی (Transfer Learning)

- پیش آموزش چند زبانه

اگر مورد استفاده هدف شامل چندین زبان باشد، مدل را روی یک مجموعه داده چند زبانه از قبل آموزش دهید. این رویکرد مشابه مدل wav2vec XLSR-53 است.

- پیش آموزش مختص به کار (Task-Specific)

مدل را بر روی وظایفی مشابه برنامه مورد نظر، مانند پیش آموزش مجموعه داده‌های مکالمه برای سیستم‌های گفتگو، از قبل آموزش دهید.

6. روش‌های Ensemble

- مجموعه مدل

خروجی‌های چند مدل wav2vec 2.0 آموزش دیده باهایپرپارامترهای مختلف یا در زیر مجموعه‌های مختلف داده را ترکیب کنید. روش‌های گروهی می‌توانند بیش از حد برازش را کاهش دهند و تعمیم را بهبود بخشند.

7. تکنیک‌های پس از پردازش

تصحیح خطا

- تکنیک‌های تصحیح خطا، مانند تصحیح املایی یا تصحیح دستور زبان را در رونویسی‌ها اعمال کنید.

امتیازدهی مجدد با Reranker

از یک مدل رتبه‌بندی مجدد برای امتیازدهی مجدد n تا بهترین فرضیه از رمزگشای جستجوی پرتو استفاده کنید، و محتمل‌ترین رونویسی را بر اساس زمینه اضافی یا قوانین زبانی انتخاب کنید.

8. تکنیک‌های آموزشی اضافی

- خود آموزی

از پیش‌بینی‌های خود مدل بر روی داده‌های بدون برچسب به‌عنوان شبه برچسب‌ها برای آموزش بیشتر مدل استفاده کنید و به طور مکرر عملکرد آن را اصلاح کنید.

- آموزش خصمانه

مثال‌های متضاد را در طول آموزش معرفی کنید تا مدل در برابر تغییرات و تحریف‌ها در ورودی قوی‌تر شود.