باسمه تعالی

دانشگاه صنعتی شریف

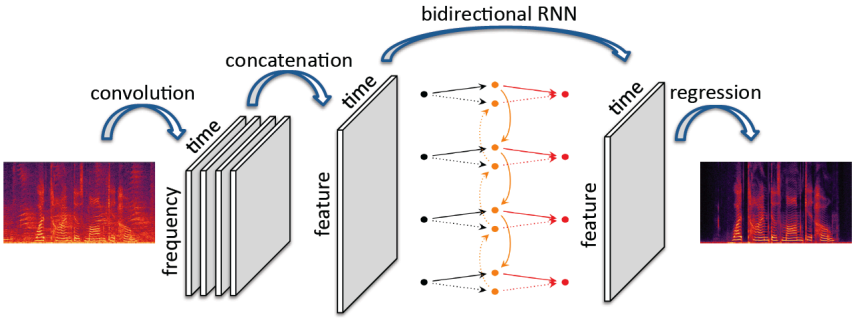
دانشکده مهندسی کامپوتیر

یادگیری عمیق

مقالات انتهایی سوال دوم

۱ سوال اول

ساختار شبکه از یک لایه کنولوشنی اولیه که کار ان بدست اوردن الگو ها در اسکتروگرام چه در حوزه زمانی و چه در حوزه فرکتنسی میباشد، در ادامه یک شبکه بازگشتی دو طرفه وجود دارد تا ارتباط فریم های متوالی را اعتبارسنجی کرده و بنسجدو در نهایت یک لایه دنس وجود دارد که اسکتوگرام خروجی را پیش بینی کند. همچنین استفاده از شبکه بازشگتی باعث شده است تا در مقایسه با دیگر شبکه های دینویز شبکه بتواند ارتباط داینامیک فریم های متوالی را چه در جهت نویزی و چه در جهت بی نویز بهتر بسنجد و پیش بینی های بهتری داشته باشد.



شکل ۱: ساختار شبکه

شبکه در ادامه فرض میکند جفت های ورودی بصرت اسکتروگرام سیگنال نویزی و اسپکتروگرام سیگنال تمیز داشته باشیم که بصورت عکس میباشند و تابع بهینه سازی را به شکل زیر تعریف میکند که پارامتر های تابع نگاشت بصورت ایتریتیو جوری تپدیت بشوند که کترین خطای تابع بهینه سازی را داشته باشیم، که این پارامتر ها از جمله وزن های کرنل های سی‌ان‌ان، وزن های شبکه بازگشتی و وزن های لایه دنس نهایی میباشند.
دقت شود شبکه وروی و خروجی کل شبکه بصورت زیر است :

$$\mathbb{R}_{+}^{d\times t} \rightarrow \mathbb{R}_{+}^{d\times t}$$

شکل ۲: ورودی و خروجیی کل شبکه

$$\min_{\theta} \quad \frac{1}{2}\sum_{i=1}^{n} ||g_\theta(\mathbf{x}_i) - \mathbf{y}_i||_F^2$$

شکل ۳: تابه بهینه سازی

سپس مقاله اشاره میکند که استفاده از چندین لایه دنس علی رغم تلاش های گذشته نمیتواند پترن های لوکالی موجود در اسپکتروگرام هارا بلدیل وجود لوکال های تکراری در بین های فرکانسی به خوبی تضخیص بدهد و لذا از کانولوشنال لایه ها استفاده میکند که بتواند این پترن هارا به خوبی تضخیص بدهد، ورودی این لایه ها همان ورودی های کل شبکه هستند که بالاتر به آن اشاره شد و در ادامه چند لایه کانولوشنی را با کرنل های مختلف که وزن های آنها جزو پارامتر های تابع هدف هستندتولید کرده و پشت هم میچیند، و از یک تابه رلو نیز بعنوان تابع فعالسازی استفاده میشود، فقط بای اینکه طول زمانی سیگنال ورودی و خروجی عینا یکسان بشود یک پدینگ صفر روی اسپکتروگرام ها قبل ورودی به لایه کانولوشنی اعمال میشود بطوریکه بعد عمودی آنهاکه به اندازه واحد های زمای است، عوض نشود و این یعنی اگر کرنل ها بعد عمودی برابر با w داشته باشند، و بعد عمودی ورودی برابر با t باشد آنگاه پدینگ به نوعی اضافه میشود که مرکز کرنل روی گوشه ای ترین پیکسل عکس ورودی بیوفتد بدین معناکه به اندازه نصف سمت راست یا چپ کرنل باید پدینگ اضافه شود که معنی آن این است که به هر طرف عکس از راست یا چپ $(w+1)/2$ پیکسل در راستای افقی به عکس اضافه میشود پس بعد عکس براب با $(d \times (w+1+t))$ میشود .

از طرف دیگر بدلیل شباهتت های فرکانسی در بینهای مجاور ، کرنل در راستای فرکانسی استرید یا مقدار حرکت برابر با نصف طول خود خواهد داشت، همچنین با نتوجه به مفروضات مقاله مشاهده میشود طول ستون های کرنل ها عددی فرد و ردیف ها عددی زوج است . که این کار باعث کاهش هزینه های محاسباتی و حافظه در جلوتر میشود.

در ادامه به علت اینکه چند فیتر مختلف گذاشتیم نتیجتا پس از لایه کانولوشنی $k(numberoflayers)$ عدد بیچر مپ $(w \times t)$ خواهیم داشت که برای اینکه بتوانی آن را به پیکه بازگشتی بدهیم بعد سوم را در بعد اول استک میکنیم تا تبدیل بشود به $R^{wk \times t}$

در نهایت هر استپ زمانی را بعنوان یک بردار R^{wk} به شبکه بازگشتی میدهیم که نتیجتا به اندازه استپ های زمانی ورودی با تعداد فیچر گفته شده خواهیم داشت و در خروجی نیز به دلیل دو طرفه بودن سیستم و اینکه رفت و برگشت را زیر هم استک میکند ، خروجی عبدی بصورت $(q \times t)$ خواهد داشت که بعد اول دو برابر حالتی است که شبکه دو طرفه نباشد، و همچنین دقت میشود که خروجی تمامی استپ های زمانی را برمیداریم که در پایتون با ارگومان $returnsequences$ تعیین میشود،

و در نهایت یک لایه دنس استفاده کرده ایم که هر استپ زمانی را به تعداد اولیه بین های فرکانسی که تعداد آنها d بود برگرداند و این یعنی ماتریس وزنی که برای لایه دنس استفاده میشود بعد $(d \times q)$ خواهد داشت ، همچنین از ترم بایاس با بعد دی نیز استفاده میشود و در نهایت بعنوان اپتیمایز و با توجه به تابع هزینه ای که در ابتدای توضیحات به آن اشاره کردیم ، از اپتیمایزر $AdaDelta$ استفاده میکنیم تا مشکلات هگرایی بوجود نیاید .

۲ سوال دوم

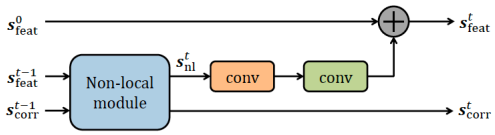
در این مقاله سعی شده است تا با استفاده از شبکه های بازشگتی و سی ان ان از ویژگی های $self - similarity$ بهره مند شده و با استفاده از متد $non - local$ به دینویز کردن عکس ها پرداخته شود.
منظور از $non - local$ در واقع این است که بچ هایی از سرتاسر عکس با توجه به میزان شباهت آنها به هم انتخاب شده و و عملگرهای مختلف را بجای اثر دادن روی کل عکس روی این بچ ها اثر میدهیم . یا بصورتی ریاضی تر و طبق گفته خود مقاله عکس هارا بعنوان ماتریس دو بعدی متشکل از پیکسل ها و چنل ها دریافت کرده و خروجی را با نگه اشتن بعد اول و تغییر دادن تعداد چنل ها تولید میکند.
یا اگر خیلی بخواهیم وارد ریاضی بشویم این بخش از مقاله این ویژگی را پوشش میدهد که فریم ورکی است که ساخته میشود و متد های نان لوکال را روی آن تاثیر میدهد .

In general, a non-local operation takes a multi-channel input $\boldsymbol{X} \in \mathbb{R}^{N\times m}$ as the image feature, and generates output feature $\boldsymbol{Z} \in \mathbb{R}^{N\times k}$. Here N and m denote the number of image pixels and data channels, respectively. We propose a general framework with the following formulation:

$$\boldsymbol{Z} = \mathrm{diag}\{\delta(\boldsymbol{X})\}^{-1}\,\Phi(\boldsymbol{X})\,\boldsymbol{G}(\boldsymbol{X})\,. \tag{1}$$

Here, $\Phi(\boldsymbol{X}) \in \mathbb{R}^{N\times N}$ is the non-local correlation matrix, and $\boldsymbol{G}(\boldsymbol{X}) \in \mathbb{R}^{N\times k}$ is the multi-channel non-local transform. Each row vector $\boldsymbol{X}_i$ denotes the local features in location i. $\Phi(\boldsymbol{X})_i^j$ represents the relationship between the $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, and each row vector $\boldsymbol{G}(\boldsymbol{X})_j$ is the embedding of $\boldsymbol{X}_j$.[1] The diagonal matrix $\mathrm{diag}\{\delta(\boldsymbol{X})\} \in \mathbb{R}^{N\times N}$ normalizes the output at each i-th pixel with normalization factor $\delta_i(\boldsymbol{X})$.

در ادامه این متد ها را روی یک شبکه بازشگتی معمولی تاثیر میدهد تا باه سختاری به شکل زیر برسد :



که بجای یک متغیر حالت در هر استپ زمانی چند متغیر حالت را تولید میکنید و که تعریف آنها را به شکل زیر انجام میدهد :

$$\boldsymbol{s}^t = f_{\text{input}}(\boldsymbol{x}^t) + f_{\text{recurrent}}(\boldsymbol{s}^{t-1}), \qquad \boldsymbol{y}^t = f_{\text{output}}(\boldsymbol{s}^t), \tag{7}$$

where f_{input}, f_{output}, and $f_{\text{recurrent}}$ are reused at every time step. In our NLRN, we set the following:

- $\boldsymbol{s}^0$ is a function of the input image $\boldsymbol{I}$.
- $\boldsymbol{x}^t = 0,\ \forall t \in \{1, \ldots, T\}$, and $f_{\text{input}}(0) = 0$.
- The output state $\boldsymbol{y}^t$ is calculated only at the time T as the final output.

و در نهایت در هر استپ زمانی یک فیچر مپ با نام s^t_{feat} و یک کاللکشن از ارتباط فیچر های عمیق ورودی با هم داریم که با s^t_{corr} نمایش داده میشود که به مجموعه این دو s^t اتلاق میشود و داریم $s^t = f_{recurrent}(s^{t-1}, s^0)$ که در تمتمی استپ های زمانی برقرار است .
روش ارایه شده در مقاله بر خلاف شبکه های بازگشتی معمولی از متد ها و فیچر های نان لوکال ورودی در شبکه بازشگتی ساتفاده میکند و متغیر های حالت جدید مطابق آنچه در بالا معرفی شد تولید میکند ،
در روش $DRCN$ یک لایه کانولوشنی را چندین بار بر روی فیچرهای ورودی اعمال میکند بدون اینکه مسیر همانی از استیت اول که بالاتر با s^0 نشان دادیم داشته باشد،
روش $DRRN$ مسیر همانی و همچنین استیت های قبلی را استفاده میکند ولیکن ا متد های نان لوکال استفاده نمیکند و نتیجه آن این است که ارتباط استیت های مجاور در ساختار شبکه جریان نمیابد و در نهایت $Memnet$ یک ارتباط دنس بین بلاک های میسازد که بلوک های مشابه وزن های شیر شده با هم دارند مادامی که بلوک های متفاوت متفاوت وزندهی میشوند، در نهایت روش مقاله در مقایسه با $MemNEt$ شبکه بازشگتی کارا تری را دارد که عمق کمتری داشته و لذا پارامتر های کمتری دارد ولی در دینویزینگ و بازسازی تصاور بهتر عمل میکند .

۳ نتیجه تمرین

مشاهده شد مدل های بازگشتی عمیق چون میتوانند همزمان بر استیت های مختلف زمانی نگاه کنند و ارتباط بین آنها را پیدا کنند لذا پتانسیل بالایی برای پیدا کردن نویز روی سیگنال های چند بعدی و یا یه بعدی و یا عکس داشته و میتوانند در صورت ست کردن درست پارامتر ها و ترکیب شدن با شبکه های عمیق دیگر و ساتفاده درست از ویژگی های سیگنال های مختلف و بویژه ویژگی های فرکانسی آنها بتوانند عمل بازسازی را انجام بدهند ولیکن اگر نویز بالا باشد همچنان قوی ترین مدل ها نیز ممکن است از بازسازی سیگنال جا مانده و نتوانند.