



پرسش های طرح شده - یادگیری عمیق (دکتر فاطمی زاده)

Composer based on RNN ۱

در این تسک، هدف پیاده سازی یک شبکه بازگشتی برای ابداع موسیقی بر اساس کد گذاری موسیقی های دلخواه است.

* در لینک زیر، کلیات کار با فرمت midi و تنسور فلو و تبدیل این فرمت به ماتریس بیان شده است. به طور کلی فرمت mid. با پخش کننده های موسیقی قابل پخش است.

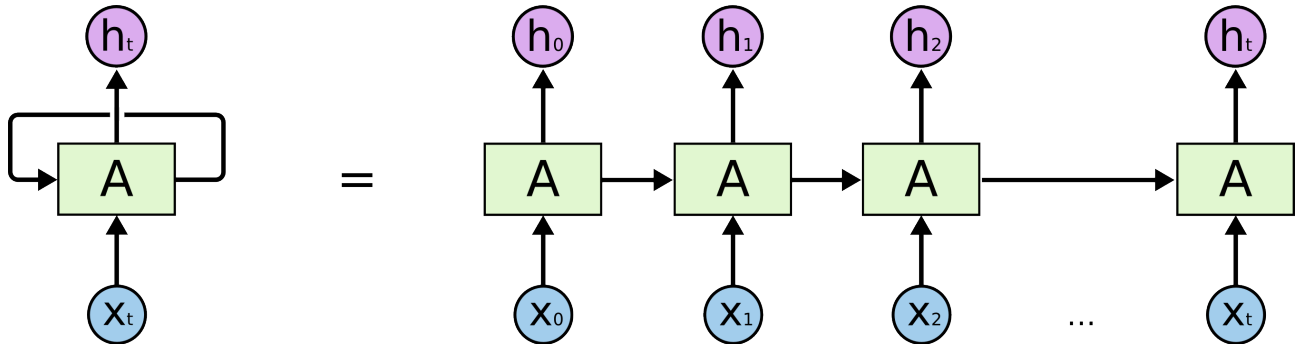
o داده های مناسب را از سایت دریافت کنید. دقت کنید در انتخاب موسیقی ها مختار هستید. برای مثال، میتوانید چندین داده برای یک ساز خاص (در مثال زیر، ساز ویولن) را برای یاددهی شبکه استفاده کنید.

```
1 !pip install bs4
2 from urllib.request import urlopen, urlretrieve
3 from bs4 import BeautifulSoup
4 import time
5 import os
6 # Define save directory.
7 save_dir = './Data/'
8 os.system('mkdir Data')
9 # Define URL components
10 url0 = 'https://www.mutopiaproject.org/cgi-bin/make-table.cgi?startat='
11 url1 = '&searchingfor=&Composer=&Instrument=Violin&Style=&collection=&id=&solo=&recent=&timelength=&timeunit=&lilyversion=&preview='
12 # Set initial values
13 songNumber = 0
14 linkCount = 10
15 lim = 10
16 cnt = 0
17 # Locate and download each MIDI file
18 while linkCount > 0 and cnt < lim:
19     cnt += 1
20     url = url0 + str(songNumber) + url1
21     html = urlopen(url)
22     soup = BeautifulSoup(html.read())
23     links = soup.find_all('a')
24     linkCount = 0
25     for link in links:
26         href = link['href']
27         if href.find('.mid') >= 0:
28             linkCount = linkCount + 1
29             urlretrieve(href, save_dir + str(cnt) + '_' + str(linkCount) + '.mid')
30     songNumber += 10
```

i توضیح دهید که چرا بهتر است موسیقی های انتخاب شده برای یاددهی شبکه، ملودی و سبک مشابه داشته باشند؟ (راهنمایی: ملودی های مشابه ویژگی معنایی مشترک بیشتری دارند).

ii به روشی داده های موسیقی mid. را کد گذاری کنید که بتوان آنها را به شبکه داد. برای مثال میتوانید از one-hot استفاده کنید. در واقع باید تابعی بنویسید که در ورودی فایل mid. را گرفته و در خروجی یک ماتریس بدهد.

iii شبکه را بر اساس کد گذاری خود طراحی کنید. ساختار کلی میتواند به صورت شکل باشد، اما در معماری آن به طور کلی سعی کنید ایده های خود را پیاده سازی کنید. شبکه باید بازگشتی باشد، اما از LSTM در این قسمت استفاده نکنید.



iv شبکه را train کنید. برای شروع، میتوانید آهنگ هایی به طول معلوم را کد گذاری کرده، و به شبکه ورودی دهید و سپس همان را در خروجی بخواهید. دقت کنید که در هنگام تست شبکه، ورودی لحظه t باید خروجی لحظه $t-1$ باشد تا شبکه به صورت متوالی کار کند. اندازه بردار حالت دلخواه است، ولی سعی کنید با توجه به پیچیدگی مورد نیاز آنرا کمتر از ۵۰ انتخاب نکنید.

v به شبکه یک حالت اولیه تصادفی (بردار حالت تصادفی) بدهید و سری ساخته شده را به فرمت midi تبدیل کنید. آنرا با استفاده از لایبری Music21 پخش کنید؛ آیا مشکل خاصی در این آهنگ وجود دارد؟

vi ممکن است شبکه ی بازگشتی شما به مشکل تشابه بیش از حد خروجی ها به ازای حالت اولیه رندم و یا تکرار تناوبی ملودی ساخته شده بر بخورد. توضیح دهید دلیل این مشکل چه چیزی در یادگیری شبکه های بازگشتی است و چگونه برطرف میشود؟

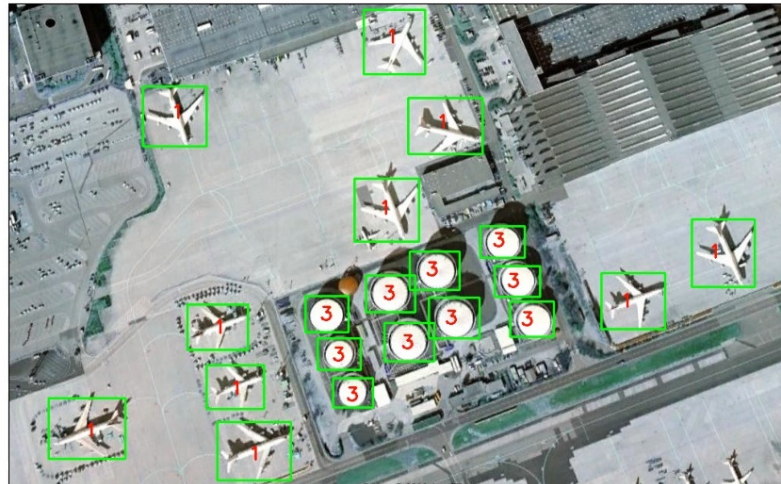
vii از بلوک LSTM استفاده کنید. نوع خروجی ها چه تغییری میکند؟ برای مقایسه با شبکه ی بازگشتی قبلی خود، دو ملودی خروجی طولانی تر از دو شبکه را مقایسه کنید.

۲ Image Object Detection

در این قسمت از تمرین می خواهیم با استفاده از شبکه های CNN، مکان و کلاس اشیا موجود در تصاویر ماهواره ای دیتاست NWPU VHR-10 را تشخیص دهیم .

* در این لینک ، دیتاست با حجم تقریبی ۷۳ مگابایت به همراه فایل راهنما قابل دریافت است . بعد از دانلود ، فایل راهنما را مطالعه و با دیتاست آشنا شوید . نمونه ای از تصویر دیتاست با نمایش bounding box و کلاس ها بصورت عکس صفحه بعد است .

o یک تابع بنام visual_dataset.py بنویسید که در ورودی نام تصویر دلخواه را بگیرد و در خروجی تصویری مانند تصویر فوق برای تصویر ورودی نمایش دهد . در گزارش خود به دلخواه چند تصویر را انتخاب و برای آن ها خروجی این تابع را نمایش دهید .

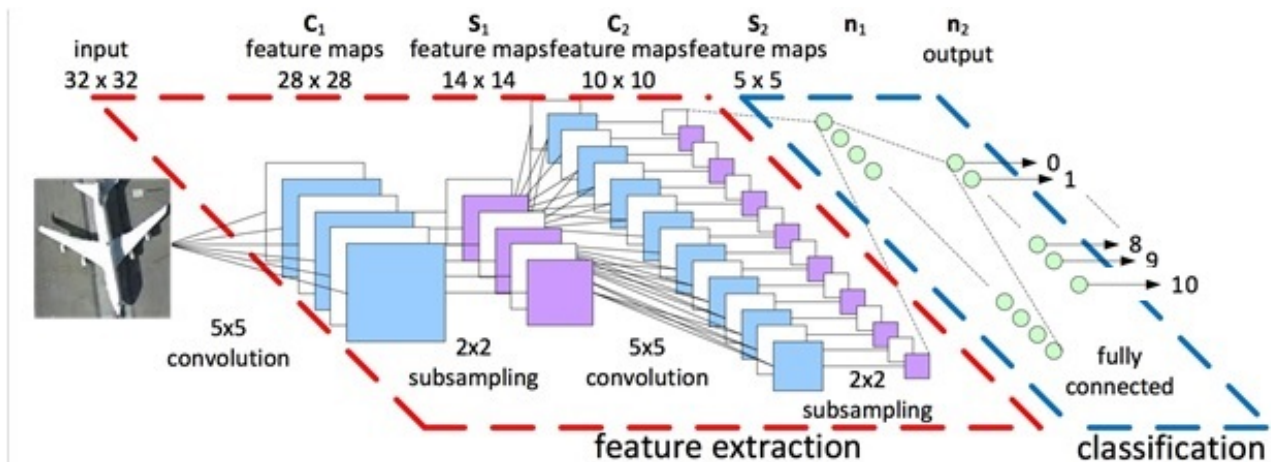


i یک تابع بنام `prepare_data.py` بنویسید که ابتدا در آن تمام `object` های ۱۰ کلاس در تمام تصاویر دیتاست را بصورت جداگانه `crop` کرده و بصورت یک لیستی از `array` ها برگرداند. همچنین کلاس متناظر با هر کدام از این `object` های `crop` شده نیز در یک لیستی دیگر نیز بعنوان خروجی دوم برگرداند.

ii همطور که از تصاویر دیتاست ملاحظه میکنید، برای کار بر روی این دیتاست یکی وجود `object` ها با سایز های متنوع و همچنین `balance` نبودن تعداد کلاس ها (این مورد را چک کنید) را باید در نظر بگیریم. چون معمولا ورودی شبکه های `cnn` که در ساختار خود `fully connected` دارند، سایز تصویر ورودی باید ثابت و فیکس باشد بنابراین در تابع قبلی تمام تصاویر `crop` شده را به سایز یکسان تبدیل کنید (به عنوان پیشنهاد 100×100). همچنین برای مشکل `unbalance` بودن داده ها تحقیق کنید که چه راه حل هایی وجود دارد و در گزارش خود توضیح دهید و از این راه حل ها برای حل این مشکل استفاده کنید. همچنین از داده های پوشه `negative`، علاوه بر داده های ۱۰ کلاس موجود در پوشه `positive`، به عنوان کلاس یازدهم استفاده میکنیم چرا که در هنگام `test` شبکه بر روی یک تصویر بزرگ (مانند تصویر فوق که چندین `object` در آن است) علاوه بر ۱۰ کلاس `positive` کلاس پس زمینه (`negative`) هم داریم که شامل هیچ کدام از ۱۰ کلاس نمی شود.

iii در این تمرین به عنوان پیشنهاد میتوانیم از یک شبکه `cnn` به عنوان `feature extraction` و سپس از یک شبکه `fully conected` به عنوان `classifier` استفاده کنیم (همانطور که احتمالا در تمرین سری قبل آشنا شده اید و می توانید از همان ایده ها نیز استفاده کنید) که در زیر شماتیکی از این شبکه را مشاهده میکنید. البته لزومی به استفاده از فقط این ساختار نیست و میتوانید ایده های خود را نیز پیاده کنید. شبکه خود را در فایل `net.py` قرار دهید. درضمن تعداد کلاس هایی که در فرایند آموزش استفاده میشوند ۱۱ تاست (۱۰ تا کلاس `positive` و ۱ کلاس `negative`) به همین دلیل نیاز است بصورت رندوم از تصاویر پوشه `negative` تصاویر `crop` شده با سایز مشابه تصاویر `positive` (به عنوان پیشنهاد 100×100) به تعداد مناسب جدا کنیم (دقت شود تعداد متناسبی جدا شود که داده های `negative` نسبت به کلاس های دیگر `unbalance` نشوند) و به داده های جمع آوری شده از `positive` ها اضافه کنیم. لیبل کلاس

negative را میتوانید ۱۱ در نظر بگیرید .



iv یک تابع بنام transform.py بنویسید که در آن augmentation های متفاوت (مانند rotation – affine – horizontalflip – verticalflip – noise – ...) وجود داشته باشد که برای فاز train استفاده خواهد شد . همچنین فقط برای داده های negative میتوانید علاوه بر augmentation های ذکر شده ، تصاویر crop شده بیشتری از پوشه negative اضافه کنید (همچنان حواستان به unbalance نشدن دیتا ها باشد) .

v ۱۰ درصد داده ها را بصورت رندوم به عنوان داده های validation انتخاب کنید و ۹۰ درصد مابقی را به عنوان داده های train در نظر بگیرید. یک تابع به نام train.py بنویسید که وظیفه آموزش شبکه را بر عهده دارد . hyper parameter های مورد نیاز را بر اساس صلاح دید خودتان ست کنید و شبکه خود را آموزش دهید. نمودار های loss و accuracy را برای هر دو داده train و validation در گزارش بیاورید . همچنین مدل آموزش دیده شده را نیز ذخیره کنید و تحویل دهید .

vi بعد از اینکه شبکه آموزش داده شد ۵ عدد از تصاویر پوشه positive را به دلخواه انتخاب کنید . برای فاز test روی تصاویر بزرگ (اصلی) انتخاب شده بدین صورت عمل میکنیم که چون سایز واقعی کلاس های positive بسیار متنوع است پس ما باید حتما شبکه را روی scale های مختلف از تصویر بزنیم که در هر scale کلاس هایی detect شود که متناسب با همان scale است . برای این کار میتوانید از تابع pyramid_gaussian داخل پکیج scikit image استفاده کنید . سپس برای هر scale به این صورت عمل میکنیم : با لغزاندن یک پنجره به ابعاد ورودی مورد نیاز شبکه روی تصویر، آن قسمت از تصویر که در زیر پنجره است را به شبکه می دهیم ، و با توجه به خروجی شبکه تصمیم گیری میکنیم که کلاس این قسمت از تصویر چند است و همین کار را انجام می دهیم تا تصویر مذکور کاملاً پوشش داده شود و به سراغ scale بعدی می رویم ، اصطلاح به این عمل sliding window میگویند . البته دقت شود بسته به نوع و ابعاد تصویر شاید نیاز نباشد پنجره یا slide های مجاور تنها در یک پیکسل اختلاف داشته باشد (که این باعث می شود فاز test بشدت کند شود) بلکه به حسب نیاز میتوان overlap را کمتر کرد (اما همچنان overlap صفر نشود) . در گزارش خود کار های انجام شده برای این قسمت را توضیح دهید و خروجی نهایی (در شکل و شمایل تصویر صفحه پیش) روی ۵ تصویر انتخاب شده را نمایش دهید . درضمن کد این قسمت را در فایل test.py قرار دهید .

vii برای بهبود نتایج قسمت قبل می توان از non-maximum-suppression(nms) و soft nms بهره



برد. در مورد این دو تحقیق کنید که چرا ممکن است به اینها نیاز پیدا کنیم و تفاوت های این دو روش نسبت به هم را بیان کنید. سپس هردو روش را روی نتایج قسمت قبل اعمال کنید (می توانید از کد های آماده برای nms استفاده کنید. کد این قسمت را در فایل nms.py قرار دهید) و نتایج هر دو روش را روی ۵ تصویر (در مجموع ۱۰ تصویر باید برای این قسمت گزارش کنید) در گزارش بیاورید.

توجه ۱: سعی کنید ۵ تصویر انتخابی را از تصاویر متنوع (سخت و آسان) انتخاب کنید. تست بر روی تصاویر حاوی فقط یک نوع کلاس یا فقط تصاویر شبیه هم (بدون تنوع) ممکن است مقداری از نمره شما را کاهش دهد.

توجه ۲: شبکه و روش شما روی چند تصویر رندوم تست خواهد شد، بنابراین قبل از ارسال کد ها از اجرا شدن بدون هیچ مشکلی اطمینان حاصل فرمایید.