

توجه : در انجام تمرین استفاده از اینترنت و مشورت مجاز میباشد اما کپی کردن تمرین حتی یک قسمت مجاز نمیباشد و در صورت مشاهده نمره کل تمرین صفر در نظر گرفته خواهد شد. برای همه تمرین نیاز است که گزارش کامل همراه کدها تحویل داده شود و گزارش نیز نمره قابل توجهی دارد. در کدها هم به میزان نیاز حتما کامنت گذاری انجام شود. در صورتی که سوالی در مورد تمرین داشتید ، ترجیحا آن را در قسمت "پرسش و پاسخ" تمرین مطرح نمایید تا سوال دانشجویان دیگر نیز برطرف شود. در صورت نیاز با ایمیل های aref.einizade@yahoo.com یا parsa.rahimi.n@gmail.com نیز می توانید مکاتبه نمایید.

اگر از IPython استفاده می کنید می توانید گزارش خود را به صورت Markdown در notebook بیاورید. در غیر این صورت گزارش خود را به صورت PDF همراه با کدهای خود ارسال نمایید.

الف) تمرین کامپیوتری :

۱) در این سوال قصد داریم با سیستم تشخیص چهره اتوماتیک opencv با استفاده از توصیف گر Haar آشنا شویم. برای این منظور ۳ فایل `haarcascade_frontalface_default.xml` (برای تشخیص چهره)، `haarcascade_eye.xml` (برای تشخیص چشم ها) و `haarcascade_smile.xml` (برای تشخیص لبخند) در اختیار شما قرار داده شده است. با استفاده از دستور `CascadeClassifier` از پکیج opencv می توانید یک `face detector` (به ازای بارگذاری فایل اول) و یک `eye detector` (به ازای بارگذاری فایل دوم) و یک `smile detector` (به ازای بارگذاری فایل سوم) بسازید. برای تشخیص صورت، چشم ها و لبخند در تصاویر تست موجود در پوشه `Test_images` نیز می توانید از متد `detectMultiScale` استفاده نمایید. تنظیمات متدها را برای تصاویر تست مختلف به گونه ای تعیین نمایید که پاسخ مطلوب بر روی تصاویر حاصل شود. کد شما باید قادر باشد بر روی تصاویر مذکور نتیجه ای مشابه زیر داشته باشد:



(۲) با استفاده از بخش data از skimage (مانند تمرین سری دوم)، سه تصویر آجر (brick)، علف (grass) و شن (gravel) را بارگذاری کنید. به عنوان مثال برای بارگذاری تصویر آجر می‌توانید به صورت زیر عمل کنید:

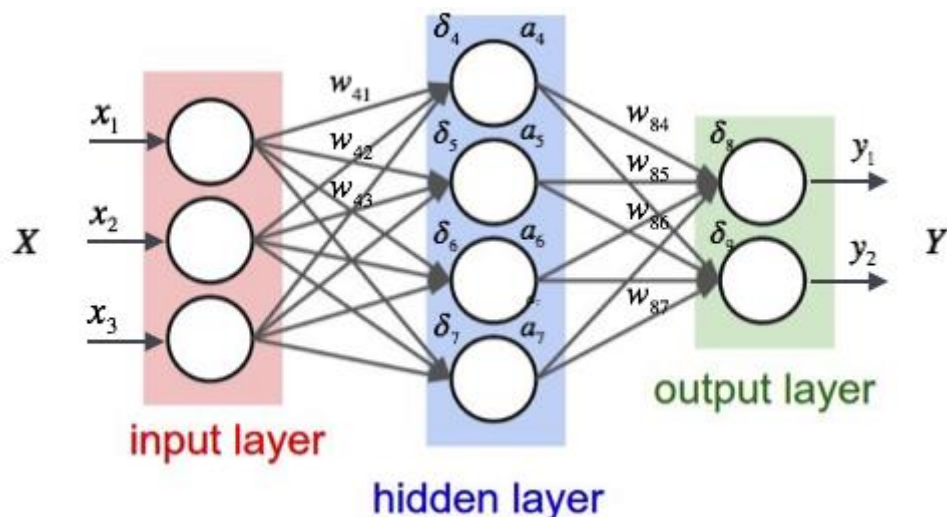
```
brick = data.brick()
```

با استفاده از تحلیل هیستوگرام و با توصیف‌گر LBP (Local Binary Pattern) توضیح دهید که چگونه می‌توان این ۳ بافت تصویری را از هم تشخیص داده و ۳ تصویر را به درستی طبقه‌بندی کرد. برای استفاده از این توصیف‌گر می‌توانید به صورت زیر عمل نمایید:

```
from skimage.feature import local_binary_pattern
```

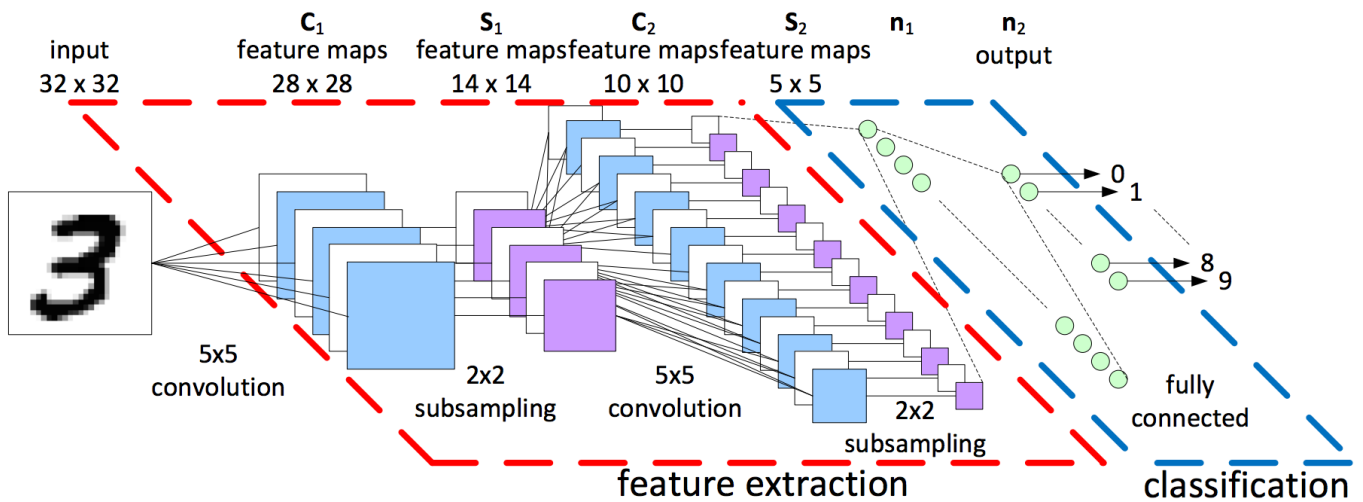
همچنین قبل از استفاده از این تابع، آرگومان‌های ورودی آن را تحلیل کنید و تنظیمات یکسانی را برای اعمال به هر ۳ تصویر در نظر بگیرید.

(۳) در ابتدا توضیح مختصری درباره شبکه‌های عصبی می‌دهیم. شبکه‌های عصبی مصنوعی در ابتدا برای مدلسازی سیستم عصبی انسان پیشنهاد شده‌اند و در ادامه به دلیل خاصیت محاسباتی آن بیش از گذشته برای حل مسایل گوناگون مورد توجه قرار گرفت. شبکه عصبی ساده تشکیل شده است از یک ورودی با ابعاد دلخواه و یک خروجی با ابعاد دلخواه متفاوت. در این بین پردازشی بر روی این داده‌ها صورت می‌پذیرد تا بتوان خروجی مطلوب تولید کرد. اگر بتوان با یک سری داده آموزشی این پردازش را یاد گرفت برای داده‌های مشابه نیز می‌توان به ازای هر ورودی خروجی مطلوب را با دقت خوبی محاسبه کرد.



شکل ۱ - شکل یک شبکه عصبی ساده

همانطور که در شکل ۱ مشاهده می‌فرمایید شبکه عصبی ساده شامل یک لایه ورودی، یک لایه خروجی و یک یا تعدادی لایه نهان است. در هرکدام از این لایه‌ها تعدادی نرون وجود دارد. هر نرون تعدادی ورودی دارد که ورودی‌ها هرکدام با خطوطی به نرون مورد نظر متصل شده‌اند. خروجی هر نرون از جمع وزن دار ورودی‌ها و سپس عبور از یک تابعی که به آن تابع فعالسازی گفته می‌شود حاصل می‌شود. یعنی ابتدا ورودی‌ها قرار گرفته و مرحله به مرحله جلو رفته و محاسبه می‌کنیم و خروجی‌ها را حساب می‌کنیم. در ابتدا وزن‌ها مقدار رندوم یا مقادیر دیگری می‌توانند داشته باشند. سپس با داده‌های آموزش که شامل ورودی‌ها و خروجی‌های نظیر هستند با محاسباتی وزن‌ها را آموزش می‌دهیم و در نهایت بعد از آموزش دیدن وزن‌ها شبکه مورد نظر بدست آمده است. برای پردازش بر روی تصاویر نیز می‌توان تصاویر را خطی کرد و به عنوان ورودی داد مثلاً یک تصویر 32×32 را به صورت یک بردار 1024 تایی به عنوان ورودی بدهیم. اما این کار خروجی‌های مطلوبی تولید نمی‌کند چون تصویر ماهیتاً دو بعدی است و دیدن پیکسل‌های کنار هم برای ما اهمیت ویژه‌ای دارد. به همین منظور شبکه‌های کانولوشنی طراحی شده‌اند. در این شبکه‌ها ورودی یک تصویر دو بعدی یا سه بعدی است که عملگر کانولوشن در هر مرحله روی آن انجام می‌شود. همچنین در هر مرحله برای کاهش بعد می‌توان pooling انجام داد و در نهایت به سبب خروجی مد نظر رسید که در صورتی که خروجی تک بعدی باشد نیاز است flatten انجام شود و خروجی نهایی مشخص شود. تصویر یک نمونه شبکه عصبی کانولوشنی را در شکل ۲ ملاحظه می‌فرمایید. در این شبکه هدف یادگیری فیلترها توسط داده‌های آموزش است که بعد از آموزش از آنها استفاده کنیم.



شکل ۲ - نمونه ای شبکه عصبی کانولوشنی برای تشخیص اعداد دست نویس

برای پیاده سازی شبکه های عصبی می توان از متلب یا پایتون بهره برد. در انجام این تمرین می توانید از کتابخانه های keras و یا tensorflow در پایتون استفاده نمایید و چون برنامه نویسی در پایتون برای شبکه عصبی و یادگیری عمیق فراگیرتر و تخصصی تر از متلب است حتما می بایست از پایتون و یکی از کتابخانه های گفته شده استفاده شود. برای گرفتن راهنمایی از طریقه استفاده از این دو کتابخانه می توانید از مثال ها و آموزش هایی که بر روی وب سایت آنها موجود است استفاده کنید و همچنین از جست و جو در اینترنت بهره ببرید.

الف) در این قسمت ابتدا نیاز است از لینک زیر از دیتاست به فرمت ۲ تصاویر مربوط به آموزش را دانلود کرده و در کد پایتون خوانده و ورودی خروجی ها را به صورت مجزا آماده کنید. لازم به ذکر است توضیحات دیتاست در سایت آن موجود است و ورودی مجموعه ای از اعداد پلاک خانه ها می باشد و خروجی می بایست ۱۰ کلاس داشته باشد یعنی در خروجی می توان ۱۰ نرون داشت که هر کدام ماکزیمم شد یعنی این تصویر ورودی عدد خروجی متناسب با آن کلاس می باشد.

ب) در قسمت بعد نیاز است که شبکه عصبی مورد نظر ما ساخته شود. پیشنهاد ما استفاده از keras به علت سهولت استفاده نسبت به tensorflow است. (پیاده سازی با tensorflow امتیاز مثبت دارد) برای ساخت شبکه این سوال چند شبکه با تعداد لایه کانولوشن متفاوت از ۲ تا ۶ و قرار دادن pooling بین آنها و همچنین در هر کدام از سائز فیلتر های ۳*۳ و ۵*۵ استفاده کنید و بعد از آموزش دقت هر کدام را بیان کرده و تفاوت آنها را بیان کنید و نتایج را تحلیل کنید. بهترین شبکه را نیز معرفی نمایید. می توانید جهت بهبود آموزش از دیتاست دیگری که درون سایت هست به نام extra در فرمت ۲ نیز استفاده نمایید و از data augmentation نیز می توانید استفاده نمایید. همچنین جست و جو کنید که dropout چیست و چه اثری بر شبکه دارد؟ آن را برای بهترین شبکه خود با درصد های متفاوت اعمال کنید و خروجی حاصل را بدست آورده و تحلیل کنید. (برای تست از داده های تست در فرمت ۲ استفاده کنید)

پ) قسمت قبل را با یک شبکه تمام متصل (با ۲ لایه تا ۶ لایه) و اعمال Dropout انجام داده و نتایج را با قسمت قبل مقایسه کنید. همچنین از نظر تعداد پارامترهای قابل یادگیری نیز این دو نوع شبکه را مقایسه کنید.

توجه: برای استفاده از شبکه‌های تمام متصل بر روی تصاویر، باید تصاویر را به بردار تبدیل کرده و سپس به ورودی این شبکه‌ها داد.

ت) همچنین برای تحلیل نتایج در قسمت (ب) می‌توانید فیلترهای میانی هر لایه را به صورت تصویر رسم کرده و با استفاده از آنها تحلیل دقیق تری ارائه نمایید. (امتیازی) همچنین تحلیل نقشه ویژگی (feature map)‌های به دست آمده از لایه اول شبکه کانولوشنی برای چند مورد از تصاویر تست نمره امتیازی دارد.

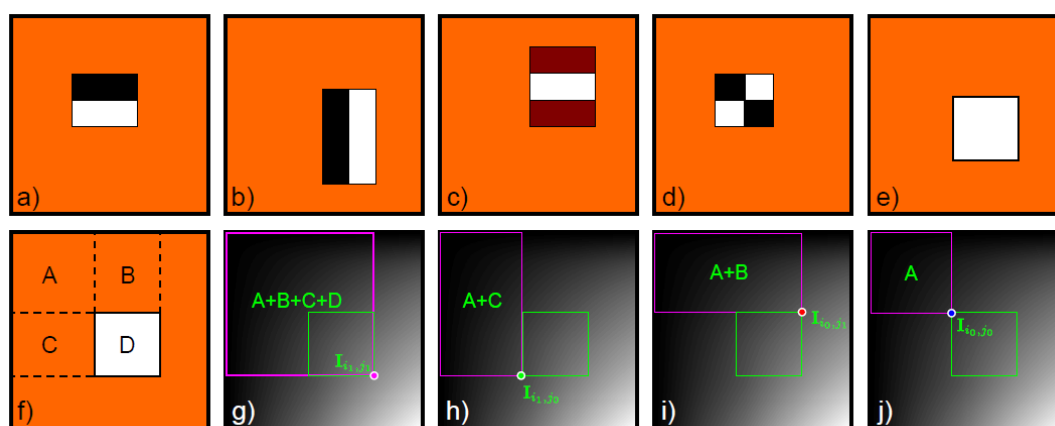
ث) حال دیتاست آموزش از فرمت ۱ را دانلود کرده، آن را خوانده و ۵ تصویر دلخواه از آن را انتخاب کرده و مشخص کنید که در کجای تصویر چه اعدادی وجود دارند. یعنی در خروجی شما می‌بایست مشخص کنید که چه اعدادی در این تصویر وجود داشتند. برای این کار می‌توانید پنجره لغزان در نظر گرفته و روی عکس بچرخانید و همچنین می‌توانید از هرم عکس تمرین قبل نیز بهره ببرید. در نهایت مشخص می‌کنید که چه اعدادی در تصویر وجود داشتند و دور هر کدام از اعداد کادر می‌کشید و عدد تشخیص داده شده نهایی توسط شبکه‌ها را در بالا و سمت چپ کادر کشیده شده توسط شبکه دور هر عدد نوشته شود. برای این منظور (مانند تمرین سری قبل) از متد Non.Maximum.Suppression استفاده نمایید. مسئله را از ابتدا تا انتها برای هر ۲ شبکه تمام متصل ساده و کانولوشنی بررسی و تفاوت نتایج را تحلیل نمایید.

لینک دیتاست مورد استفاده در این تمرین :

<http://ufldl.stanford.edu/housenumbers/>

ب) تمرین تئوری:

۱) مجموعه‌ای از عملیات‌هایی که نیاز است انجام شود تا Haar-like فیلترها در شکل زیر از یک integral image بدست آیند را شرح دهید. برای محاسبه هر کدام از integral image چه تعداد نقطه نیاز است؟



۲) در مورد روش SIFT به سوالات زیر پاسخ دهید.

الف) می دانیم تعداد بینهای هیستوگرام در مرحله انتساب جهت و ساخت توصیفگر متفاوت است. چه توجیهی برای این تفاوت می توان بیان کرد؟

ب) در مرحله ساخت توصیفگر، توضیح دهید چگونه برای کاهش محاسبات از چرخاندن مستقیم پنجره 16×16 حول نقطه کلیدی اجتناب می شود.

۳) به سوالات زیر پاسخ دهید.

الف) آیا گوشه یاب هریس یک فیلتر خطی است؟ چرا؟

ب) فرض کنید گوشه یاب هریس به یک تصویر smooth نشده اعمال شده است. چه نوع تصویری گوشه یاب را در جاهایی که به وضوح گوشه ندارند دچار خطا می کند؟

پ) تمرین پژوهشی:

با مراجعه به این مقاله توضیح دهید که Image Inpainting چیست و برای رسیدن به مقصود این تکنیک چه روش های وجود دارد. (۱ تا ۲ مقاله مشابه به مقاله ذکر شده را مطالعه نمایید و روش استفاده شده را به طور خلاصه شرح دهید. سعی کنید که از مقالات جدید استفاده کنید برای این کار بهتر است کنفرانس های معتبر Computer Vision مانند CVPR, ICCV, ECCV , در سال های اخیر بررسی شود)