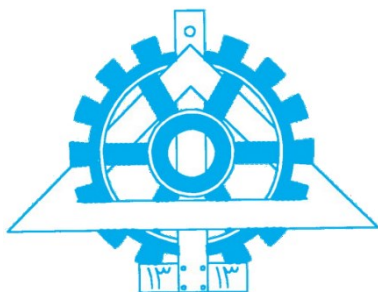


به نام خدا

دانشکده مهندسی برق و کامپیوتر

درس یادگیری عمیق

تمرین سری دوم



در این تمرین هدف پیاده سازی شبکه ی u-net به منظور انجام semantic segmentation برای دیتاست CamVid می باشد. در ابتدا شبکه اصلی را پیاده سازی کرده و نتایج آن را بررسی میکنیم. سپس نتیجه ی انجام تغییراتی در ساختار شبکه را مورد بررسی قرار میدهم. در بخش دوم تمرین تلاش میکنیم با استفاده از transfer learning از طریق شبکه semantic segmentation یک شی را به یک عکس اضافه کنیم.

در تمامی بخش های این تمرین از دیتاست CamVid که آدرس آن در زیر آورده شده است استفاده کنید.

<https://github.com/mostafaizz/camvid>

بخش اول

منظور از semantic segmentation استفاده از یک طبقه بند برای اختصاص دادن هر یک از پیکسل های یک عکس به یک کلاس مشخص می باشد. از طریق این الگوریتم با توجه به موقعیت مکانی و ارتباط هر پیکسل با پیکسل های کناری برای هر یک از کلاس ها loss مربوط به آن کلاس محاسبه گردیده و بر این اساس label پیکسل مربوطه تعیین می شود.

شبکه u-net شبکه ای است که به منظور انجام semantic segmentation بر روی داده های biology توسعه یافته است. این شبکه با استفاده از لایه های کانولوشن و سپس استفاده از up sampling ویژگی های مربوط به context عکس را استخراج کرده و با در کنار هم قرار دادن این ویژگی ها و feature map ها استخراج شده در هر مرحله اطلاعات local و context را ترکیب میکند. توضیحات تکمیلی این شبکه در مقاله زیر قابل مشاهده می باشد.

<https://arxiv.org/pdf/1505.04597.pdf>

همچنین لازم به ذکر است که در قسمت مربوط به up sampling از الگوریتم نزدیکترین همسایه (Nearest Neighbor) استفاده نماییم. همچنین از cross entropy به عنوان تابع loss استفاده نماییم.

توجه: تغییرات هر قسمت را نسبت به شبکه اصلی ذکر شده در بخش الف انجام دهید و تغییرات مربوط به هر قسمت را به قسمت بعد انتقال ندهید.

- I. شبکه u-net را مطابق با آنچه در مقاله ذکر شده است پیاده سازی کرده و با استفاده از روش Adam آموزش دهید. نمودار loss داده آموزش و تست را در حین آموزش نمایش دهید. یک عکس از مجموعه عکس های داده تست را به شبکه وارد کرده و خروجی آن را به همراه عکس ورودی در گزارش خود نمایش دهید. همچنین تعداد اپیک لازم برای آموزش مناسب شبکه را گزارش کنید.
- II. در تمامی بخش های شبکه دو لایه کانولوشن متوالی $3 * 3$ را با یک لایه کانولوشن $5 * 5$ جایگزین کرده و آموزش را تکرار کرده و نمودار loss داده آموزش و تست را در حین آموزش نمایش دهید. همچنین نتایج و تعداد پارامترهای این دو شبکه را مقایسه کنید.
- III. روش بهینه سازی را به SGD تغییر داده و نمودار loss داده آموزش و تست در حین آموزش را نمایش داده و با شبکه اصلی مقایسه کنید.
- IV. در تمامی بخش های شبکه هر لایه کانولوشن و max pool متوالی را با یک لایه کانولوشن با $\text{stride}=2$ جایگزین کنید و نمودار ذکر شده در بخش های قبل را رسم کرده و با نتایج شبکه اصلی مقایسه کنید.
- V. در لایه های کانولوشنی که بعد از آنها عملیات max pooling انجام میشود، پیش از تابع غیرخطی ReLU عملیات Batch Normalization را انجام دهید و نتایج را با نتیجه قسمت الف مقایسه کنید.

بخش دوم

در این بخش از یک مدل از قبل آموزش دیده (pre trained) برای semantic segmentation استفاده میکنیم و تلاش میکنیم این شبکه را برای داده خود fine-tune کرده و در نهایت از این شبکه برای افزودن یک شی (ماشین، انسان و ...) به یک تصویر استفاده میکنیم. (از مدل قسمت قبل در این بخش استفاده ننمایید.)

برای مدل از قبل آموزش داده شده میتواند از پروژه موجود در آدرس زیر استفاده نمایید.

<https://github.com/CSAILVision/semantic-segmentation-pytorch>

در مدل مورد نظر انکودر را ResNet50dilated و دیکودر را PPM deep sup (کد مربوط به این مدل در پروژه مورد نظر موجود می‌باشد) در نظر بگیرید. برای دانلود مدل از قبل آموزش دیده میتوانید از فایل demo_test.py موجود در پروژه استفاده نمایید.

پس از دانلود مدل برای یک ایپاک از داده خود برای آموزش شبکه از قبل آموزش دیده استفاده کنید و از این طریق وزن‌های شبکه را تنظیم کنید. به این کار fine-tune کردن شبکه‌ی pre trained گفته می‌شود.

پس از fine-tune کردن شبکه، یک عکس از مجموع عکس‌های تست را انتخاب کرده و عکس را از شبکه عبور دهید. سپس در وسط خیابان به خروجی شبکه یک کلاس (مثلاً اتومبیل، انسان و ...) اضافه نمایید. افزودن کلاس را می‌توانید از طریق تغییر مقدار آن کلاس در خروجی شبکه انجام دهید. حال با چند بار انجام back propagation عکس ورودی را به روزرسانی کنید. دقت کنید وزن‌های شبکه در این بخش ثابت می‌باشند. در نهایت عکس تغییر یافته را به همراه خروجی شبکه برای آن (عکس segment شده) در گزارش خود قرار دهید.

همچنین در هر مرحله یکی از فیلترهای لایه اول را به صورت یک عکس visualize کنید.

نکات:

- توجه کنید که نیمی از نمره تمرین مربوط به گزارش می‌باشد. لازم به ذکر است که رعایت اصول نگارشی حائز اهمیت می‌باشد.
- در صورتی که امکان اجرای کد بر روی سیستم خود را ندارید می‌توانید از google colab استفاده نمایید.
- در صورتی وجود هرگونه اشکال در اجرای کد نمره صفر برای این تمرین لحاظ خواهد شد. در صورتی که از jupyter notebook استفاده می‌کنید دقت کنید که کد شما به صورت cell به cell اجرا شود.
- گزارش تمرین را حتماً به صورت PDF ارسال نمایید.
- کدهای تمرین را به همراه گزارش تمرین در سایت درس آپلود نمایید.
- نحوه نام‌گذاری تمرین براساس studentnumber_homeworknumber.pdf باشد.
- زبان پیاده‌سازی این تمرین python می‌باشد.
- هر گونه پرسش پیرامون تمرین را با ایمیل‌های sepehr.sameni@gmail.com و raminnakhli@gmail.com و esmaeilfarahng@gmail.com مکاتبه فرمایید.