به نام خدا

یادگیری عمیق

پاییز ۱۴۰۳

استاد: دکتر فاطمی‌زاده

دانشگاه صنعتی شریف
دانشکده‌ی مهندسی برق

گردآورندگان: سجاد هاشم بیکی، محمدحسین فرامرزی، محمدحسین عاشوری

تمرین سوم | **شبکه‌های عمیق کانولوشنی** | مهلت ارسال: جمعه ۱۶ آذر

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص‌شده است.
- در طول ترم امکان ارسال با تاخیر پاسخ همه‌ی تمارین تا سقف ۵ روز و در مجموع ۱۵ روز، وجود دارد. پس از گذشت این مدت، پاسخ‌های ارسال‌شده پذیرفته نخواهند بود. همچنین، به ازای هر روز تأخیر غیر مجاز ۱۰ درصد از نمره تمرین به صورت ساعتی کسر خواهد شد.
- همکاری و همفکری شما در انجام تمرین مانعی ندارد اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد. (دقت کنید در صورت تشخیص مشابهت غیرعادی برخورد جدی صورت خواهد گرفت.)
- در صورت همفکری و یا استفاده از هر منابع خارج درسی، نام همفکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.
- نتایج و پاسخ های خود را در یک فایل با فرمت zip به نام HW۳-Name-StudentNumber در سایت CW قرار دهید. برای بخش عملی تمرین نیز در صورتی که کد تمرین و نتایج خود را در گیت‌هاب بارگذاری می‌کنید، لینک مخزن مربوطه (repository) را در پاسخنامه خود قرار بدهید. دقت کنید هر سه فایل نوتبوک تکمیل شده بخش عملی را در گیت‌هاب قرار دهید. همچنین لازم است تا دسترسی های لازم را به دستیاران آموزشی مربوط به این تمرین بدهید.
- لطفا تمامی سوالات خود را از طریق صفحه درس در سایت Quera مطرح کنید (برای اینکه تمامی دانشجویان به پاسخ‌های مطرح شده به سوالات دسترسی داشته باشند و جلوی سوالات تکراری گرفته شود، به سوالات در بسترهای دیگر پاسخ داده نخواهد شد).
- دقت کنید کدهای شما باید قابلیت اجرای دوباره داشته باشند، در صورت دادن خطا هنگام اجرای کدتان، حتی اگه خطا بدلیل اشتباه تایپی باشد، نمره صفر به آن بخش تعلق خواهد گرفت.

سوالات نظری (۱۰۰ نمره)

۱. (۲۵ نمره) در این سوال مباحث محاسباتی مربوط به شبکه‌های کانولوشنی را بررسی می‌کنیم.

(آ) یک لایه کانولوشن معمولی با کرنل $K \times K$ را با فرض Same Padding و Stride $= ۱$ به یک Feature Map با ابعاد $H \times W \times M$ اعمال می‌کنیم که شامل M کانال است. تعداد کانال‌ها در خروجی این لایه را برابر با N در نظر بگیرید. تعداد پارامترها و هزینه محاسباتی (تعداد عملیات ضرب لازم) این لایه کانولوشن را برحسب متغیرهای H, W, M, N, K بدست آورید.

(ب) حالا یک شبکه کانولوشنی را در نظر بگیرید که ورودی آن یک تصویر رنگی با ابعاد ۱۲۸ $\times$ ۱۲۸ است. فرض کنید شبکه دارای سه لایه کانولوشن متوالی با کرنل‌های ۵ $\times$ ۵ و Padding $= ۲$ و Stride $= ۲$ با تابع فعال‌سازی ReLU است. لایه‌های کانولوشن به ترتیب دارای ۶۴، ۱۲۸ و ۲۵۶ فیلتر هستند. به موارد زیر پاسخ دهید.

- تعداد پارامترها، هزینه محاسباتی و ابعاد خروجی در هر لایه کانولوشن را محاسبه کنید.
- Receptive field را برای یک نورون از آخرین لایه کانولوشن بررسی کنید. به بیان دیگر، هر یک از نورون‌های آخرین لایه کانولوشن از چه تعداد از پیکسل‌های تصویر ورودی تأثیر می‌پذیرد؟

(ج) در این بخش به مبحث کانولوشن‌های جداشدنی عمقی (Depthwise Separable Convolutions) می‌پردازیم که در معماری شبکه MobileNet بکار رفته است. به موارد زیر پاسخ دهید.

- تعداد پارامترها و هزینه محاسباتی برای یک لایه کانولوشن جداشدنی عمقی را بدست آورید و با کانولوشن معمولی (بخش آ) مقایسه کنید.
- شبکه کانولوشنی بخش ب را مجدداً در نظر بگیرید. اما حالا فرض کنید دومین و سومین لایه‌های کانولوشن این شبکه را مانند MobileNet با کانولوشن‌های جداشدنی عمقی جایگزین کنیم. تعداد پارامترها و هزینه محاسباتی لایه‌های کانولوشنی را با بخش ب مقایسه کنید.

(د) فرض کنید یک مسئله طبقه‌بندی با ۲۰۰ کلاس را بررسی می‌کنیم. برای این منظور لایه Flatten به همراه یک لایه Fully Connected با توابع فعالسازی SoftMax را به لایه‌های کانولوشنی اضافه می‌کنیم. مجموع تعداد پارامترهای این شبکه را برای معماری‌های کانولوشنی بخش‌های ب و ج محاسبه و با یکدیگر مقایسه کنید. سهم لایه Fully Connected از تعداد پارامترها چقدر است؟ برای کاهش تعداد پارامترها راهکار(هایی) را پیشنهاد دهید و تأثیر آن را بررسی کنید.

۲. (۲۵ نمره) در این تمرین قصد داریم به بررسی Densely Connected Convolutional Networks بپردازیم. برای مطالعه این شبکه می‌توانید به لینک زیر مراجعه نمایید.

- Densely Connected Convolutional Networks

- تفاوت‌های اصلی DenseNet's dense connections و ResNet's residual connections را بیان کنید. در مورد هر کدام از موارد گفته شده نیز توضیح مختصری بدهید.
- بیان کنید که DenseNet چگونه مشکل vanishing gradient را کاهش می‌دهد و مزیت محاسباتی آن چه می‌باشد؟
- چه زمانی استفاده از DenseNet در یک مسئله عملی پیشنهاد می‌شود؟ یک مثال واقعی بیاورید که این معماری برای آن مناسب باشد.
- اگر داده‌های ورودی متشکل از چند modality (مانند تصویر و متن) باشند، چگونه می‌توان DenseNet را برای پردازش این داده‌ها تطبیق داد؟ ساختار پیشنهادی خود را رسم کنید و توجیه کنید.

۳. (۲۵ نمره)

در درس با ساختار شبکه U-Net آشنا شدیم. در این سوال قصد داریم در ابتدا ویژگی های اصلی معماری و آموزش این شبکه را بررسی کنیم و در نهایت به عملگر Transposed Convolution بپردازیم. در تصویر زیر نمایی کلی از معماری شبکه را مشاهده میکنید. برای اشنایی بیشتر توصیه میشود مقاله مربوطه را مطالعه کنید.

(آ) در این شبکه، دو بخش انکودر و دیکودر با استفاده از Skip Connection با یکدیگر ارتباط دارند. دلیل و تاثیر وجود این اتصالات را با توجه به مطالب مطرح شده در مقاله شرح دهید.

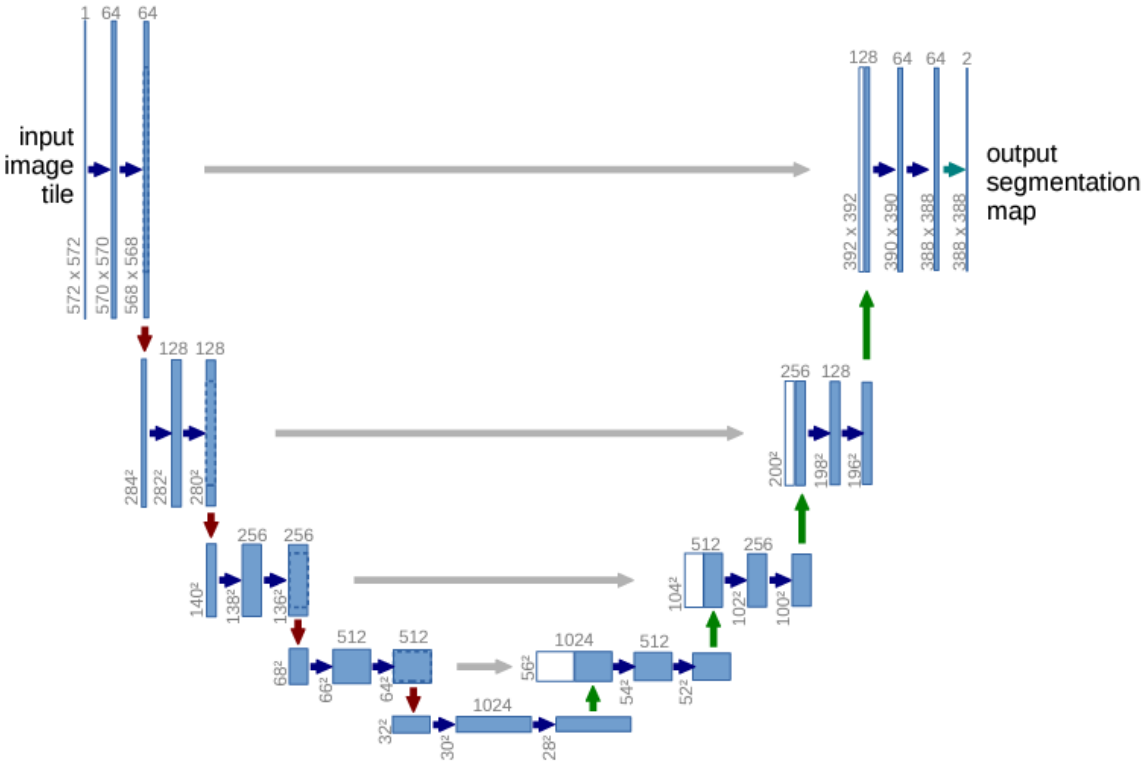
(ب) برای آموزش شبکه، از تکنیک Random Deformation برای افزایش تعداد داده های آموزشی استفاده شده است. باتوجه به مطالب بیان شده در مقاله، چگونگی انجام این تکنیک را توضیح داده و تاثیر وجود آن را در عملکرد مدل بیان کنید.

(ج) دو ماتریس زیر را در نظر بگیرید.با استفاده از فیلتر و ورودی تعیین شده، عملگر Transposed Convolution را اعمال کنید.

$$\text{ورودی} = \begin{pmatrix} ۱ & ۲ \\ ۳ & ۴ \end{pmatrix}$$

$$\text{فیلتر} = \begin{pmatrix} ۱ & ۲ \\ ۳ & ۴ \end{pmatrix}$$

توجه: در بخش (ج) بایستی مراحل انجام به طور کامل شرح داده شود، صرفاً برای چک کردن جواب نهایی میتوانید از کتابخانه پایتورچ استفاده کنید.



۴. (۲۵ نمره) در زمینه تشخیص اشیا، الگوریتم‌های کارآمد و یکپارچه YOLO (You Only Look Once) بر مبنای پردازش یکباره تمام تصویر طراحی شده اند و به دلیل امکان تشخیص بی‌درنگ (Real-time) و دقیق اشیا، به طور گسترده در پروژه‌های دنیای واقعی استفاده می‌شوند. نسخه‌های پایه‌ای اول تا سوم YOLO را در این تمرین بررسی می‌کنیم. جهت مطالعه مقالات می‌توانید به لینک‌های زیر مراجعه کنید.

- You Only Look Once: Unified, Real-Time Object Detection
- YOLO9000: Better, Faster, Stronger
- YOLOv3: An Incremental Improvement

(آ) فرض کنید از یک دیتاست تشخیص اشیا شامل ۸۰ کلاس استفاده می‌کنیم. تعداد کانالها (عمق) در خروجی شبکه‌های YOLOv1 و YOLOv3 را با یکدیگر مقایسه کنید و دلیل تفاوت را ذکر کنید. (تعداد Bounding box در هر سلول را مطابق مقالات اصلی در نظر بگیرید.)

(ب) ممکن است نمونه‌های دیتاست از نظر برچسب دارای همپوشانی باشند و هر شی دقیقاً به یک کلاس دیتاست تعلق نداشته باشد. چه راهکاری در YOLOv3 برای غلبه به این مشکل ارائه شده است؟

(ج) در مقالات YOLO، برای جلوگیری از تشخیص تکراری و چندگانه اشیا چه الگوریتمی استفاده شده است؟

(د) توضیح دهید که چرا برخلاف YOLOv1 در YOLOv2 و YOLOv3 رویکرد آموزش شبکه بر روی تصاویری با اندازه‌های مختلف امکان پذیر است؟ این ایده چگونه پیاده‌سازی شده است و چرا سودمند است؟

(ه) در مقاله YOLOv2 دو مشکل اصلی برای استفاده از Anchor box با YOLO ذکر شده است. این مشکلات را بیان کنید و راهکارهایی که مقاله YOLOv2 ارائه کرده است را توضیح دهید.

(و) معماری شبکه YOLOv3 چه تفاوت‌های کلیدی نسبت به قبل دارد؟

سوالات عملی (۳۰۰ نمره)

۱. (۱۰۰ نمره)

در این سوال میخواهیم یک شبکه CNN پایه برای طبقه بندی تصاویر طراحی کنیم و اثر لایه های مختلف در شبکه را بررسی کنیم.

(آ) دیتاست مورد استفاده در این تمرین CIFAR-10 میباشد که شامل در مجموع ۶۰۰۰۰ تصویر، مشتمل بر ۵۰۰۰۰ تصویر آموزشی و ۱۰۰۰۰ تصویر آزمون در ۱۰ دسته مختلف است. ابتدا با استفاده از torchvision.datasets دیتاست CIFAR-10 را دانلود و داده های train و test جدا کنید. سپس ۱۰۰۰۰ تصویر از داده های train را برای validation جدا کنید. در صورت نیاز در هنگام دانلود پیش پردازش های مورد نیاز را نیز روی داده ها انجام دهید. از هر کلاس موجود در دیتاست یک تصویر رندوم به همراه لیبل آن نمایش دهید.

(ب) با استفاده از torch.utils.data.DataLoader داده های خود را به batch های مختلف تقسیم کنید. سپس شبکه از پیش نوشته شده زیر را آموزش دهید و accuracy و loss را بر روی داده های train و validation را گزارش کنید.

```
class BaselineModel(nn.Module):
    def __init__(self):
        super(BaselineModel, self).__init__()
        self.conv1 = nn.Conv2d(in_channels=3, out_channels=32, kernel_size=5)
        self.relu = nn.ReLU()
        self.maxpool = nn.MaxPool2d(kernel_size=2, stride=2)
        self.flatten = nn.Flatten()
        self.fc = nn.Linear(32 * 14 * 14, 10)

    def forward(self, x):
        x = self.conv1(x)
        x = self.relu(x)
        x = self.maxpool(x)
        x = self.flatten(x)
        x = self.fc(x)
        return x


baseline_model = BaselineModel().to(device)
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD()
```

اندازه batch و تعداد epoch ها را خودتان تعیین کنید (حداکثر ۳۰ ایپاک). نمودار loss و accuracy در حین آموزش را بر حسب epoch رسم کنید. همچنین با استفاده از دستور torch.save بهترین مدل این بخش را ذخیره کنید.

(ج) شبکه baseline را ارتقا دهید. برای ساخت شبکه عصبی مورد نظر از حداکثر ۴ لایه کانولوشنی به همراه تعدادی لایه pooling استفاده کنید. انتخاب سایز فیلتر ها و تعداد آن ها در هر لایه بر عهده شما است. حداکثر تعداد لایه های FC نیز ۳ میباشد. نتایج را گزارش کنید.

(د) در درس با لایه Batch Normalization آشنا شدید. لایه (لایه های) BN را به صورت مناسب در بین لایه های کانولوشنی بهترین شبکه قسمت قبل قرار داده و نتایج را گزارش و تحلیل کنید.

(ه) به بهترین مدل قسمت قبل این بار لایه (لایه های) Dropout در میان لایه های FC اضافه کرده و مجددا نتایج را گزارش کنید. آیا نتایج بهبود یافته اند؟ تحلیل کنید.

(و) در نهایت با استفاده از بهترین مدل هر کدام از بخش های (ب) تا (ه) که قبلا ذخیره کرده اید ، ۱۰۰۰۰ تصویر test را طبقه بندی کنید ، دقت و confusion matrix را گزارش دهید.

توجه: در هرکدام از بخش های (ج) تا (ه) مانند بخش (ب)، مقدار loss و accuracy را در انتهای هر epoch بر روی داده های train و validation گزارش کنید. درنهایت نمودار loss و accuracy بر حسب epoch را رسم کنید و بهترین مدل هر بخش را نیز ذخیره کنید.

۲. (۱۰۰ نمره) هدف از این سوال آشنایی و بررسی معیار های loss مختلف و تاثیر آنها بر فرآیند آموزش یک شبکه کانولوشنی می باشد. نکته‌ای که وجود دارد این است که لزوما این تابع ضرر، مختص به لایه آخر شبکه عصبی نیست و میشود از آن درلایه‌های میانی نیز استفاده کنیم. همچنین، در بسیاری از موارد برای بهبود آموزش، می توانیم از جمع وزندار چند تابع ضرر به صورت همزمان استفاده کنیم.

برای این سوال، از دیتاست cifar10 استفاده می کنیم. این مجموعه داده را از لینک زیر می توانید دانلود کنید:

Cifar10 dataset

هدف ما آموزش یک classifier برای دو کلاس هواپیما (airplane) و ماشین (automobile) از دیتاست CIFAR-10 است.

بخش یک - آموزش مدل با معیار Cross Entropy Loss

(آ) یک شبکه کانولوشنی ساده یا از پیش‌آموزش‌دادەشده (مانند ResNet-50 یا VGGNet) استفاده کنید. همانطور که می‌دانیم، این نوع شبکه‌ها، شامل یک لایه fully connected هستند که ورودی آن برداری به اندازه feature vector استخراج‌شده و خروجی آن به اندازه تعداد کلاس‌ها است. دقت کنید این کلاس ها، دو دسته هواپیما (airplane) و ماشین (automobile) از دیتاست CIFAR-10 هستند.

(ب) مدل را با استفاده از معیار Cross Entropy Loss برای طبقه‌بندی داده‌ها آموزش دهید.

(ج) نمودار دقت (Accuracy) و خطا (Loss) را برای هر epoch رسم کنید.

(د) نقشه ویژگی (Feature Map) را در طول فرآیند آموزش برای چند لایه اصلی از شبکه استخراج و نمایش دهید. این نقشه نشان می‌دهند که چگونه مدل، ویژگی‌های مختلف را در هر لایه شبکه شناسایی و پردازش می‌کند. برای این کار، می‌توانید داده ورودی را از میان نمونه‌های آموزشی انتخاب کنید و خروجی لایه‌های کانولوشنی را تصویرسازی نمایید.

بخش دوم - آموزش مدل با معیار Triplet Loss

این بار به جای استفاده از cross entropy loss از triplet loss استفاده کنید. (برای آشنایی بیشتر با triplet loss می‌توانید از این لینک استفاده کنید.) به موارد زیر دقت کنید:

(آ) به دلیل ماهیت triplet loss کلاس مربوط به دیتاست باید توسط خودتان نوشته شود.

(ب) هدف در این قسمت این است که با استفاده از triplet loss ابتدا یک feature extractor خوب آموزش دهیم (دقت شود در این بخش آموزش، لایه fully connected دخیل نشده). پس از آموزش داده شدن feature extractor، حال لایه fully connected را با فریز کردن وزن لایه های قبلی، با cross entropy loss آموزش دهید.

(ج) نمودارهای خواسته شده در بخش اول را برای این بخش نیز رسم کنید (هم برای آموزش feature extractor و هم برای آموزش لایه classifier) و دقت نهایی این مدل را بر روی دیتاست تست حساب کنید.

(د) تغییرات نقشه ویژگی (Feature Map) در مقایسه با مرحله قبلی را بررسی و تحلیل کنید.

بخش سوم - مقایسه و نتیجه گیری

با استفاده از دو معیار Cross Entropy Loss و Triplet Loss در مراحل مختلف آموزش شبکه کانولوشنی (CNN)، تحلیل کنید که هر یک از این معیارها چگونه بر عملکرد مدل در زمینه‌های زیر تأثیر می‌گذارند:

- کیفیت نقشه ویژگی (Feature Map) در لایه‌های مختلف شبکه
- دقت نهایی مدل بر روی داده‌های تست
- سرعت همگرایی (Convergence) در طول فرآیند آموزش

در نهایت، نتیجه‌گیری کنید که کدام معیار برای کاربردهای مختلف، از جمله قابلیت تعمیم دادن به داده های جدید و همچنین طبقه‌بندی داده‌های پیچیده و استخراج ویژگی‌های متمایز، مناسب‌تر است.

بخش چهارم - ترکیب توابع معیار های loss

این بار می‌خواهیم این دو تابع loss را همزمان در آموزش دخیل کنیم. برای اینکار، تابع ضرر زیر را در نظر بگیرید:

$$L_{\text{total}} = L_{\text{triplet}} + L_{\text{cross-entropy}}$$

عمل backpropagation را بر روی L_{total} انجام دهید. به موارد زیر دقت کنید:

(آ) برخلاف قسمت قبل، که ابتدا مدل استخراج ویژگی آموزش داده می‌شد و سپس classifier، در اینجا کل مدل در حال‌ترین شدن است.

(ب) نمودارهای خواسته شده را در این بخش نیز رسم کنید.

(ج) دقت نهایی این مدل را بر روی دیتاست تست حساب کنید.

(د) در مورد تاثیر این ترکیب معیار ها و علت آن تحلیل خود را گزارش کنید.

۳. (۱۰۰ نمره) در این سوال با استفاده از Transfer Learning یک مسئله طبقه‌بندی را بررسی می‌کنیم. برای استخراج ویژگی می‌توان از شبکه‌های کانولوشنی معروف که قبلاً بر روی دیتاست وسیع ImageNet آموزش دیده اند استفاده کرد که از Torchvision قابل دریافت هستند.

(آ) با استفاده از Torchvision، شبکه MobileNetV2 را به همراه وزن‌های از قبل آموزش‌دیده پیاده‌سازی کنید. مختصراً معماری این شبکه، ابعاد ورودی، پیش‌پردازش‌های لازم برای ورودی و ابعاد خروجی آن را توضیح دهید.

(ب) یک تصویر رنگی با کیفیت مناسب را در نظر بگیرید به نحوی که از کلاس‌های قابل تشخیص توسط شبکه باشد. این عکس را پیش‌پردازش کنید و خروجی شبکه MobileNetV2 را بدست آورید. نام‌های سه کلاس پیش‌بینی‌شده با بیشترین احتمال را مشخص کنید.

(ج) از دیتاست Oxford 102 Flower برای طبقه‌بندی استفاده می‌کنید که شامل ۸۱۸۹ تصویر از ۱۰۲ دسته متفاوت از گل‌ها است. برای توضیحات بیشتر در خصوص دیتاست می‌توانید به لینک اول و لینک دوم مراجعه کنید. تصویر ورودی را به ابعاد ۲۲۴ × ۲۲۴ پیکسل در نظر بگیرید و از شبکه MobileNetV2 برای استخراج ویژگی استفاده کنید و لایه‌های مربوط به طبقه‌بندی را اضافه کنید. با استفاده از تابع هزینه Cross Entropy شبکه حاصل را برای طبقه‌بندی ۱۰۲ کلاس دیتاست آموزش دهید. در این بخش فرض کنید که وزن‌های بخش استخراج ویژگی بصورت از قبل آموزش‌دیده و Freeze هستند. تنها بخش طبقه‌بندی آموزش داده می‌شود که لازم است مقداردهی اولیه شود.

خواسته‌های زیر را گزارش کنید:

- نمودار تغییرات دقت و هزینه را برای داده‌های آموزشی و ارزیابی برحسب Epoch رسم کنید.
- پس از تکمیل آموزش شبکه، دقت و هزینه را برای داده‌های آموزشی، ارزیابی و تست محاسبه کنید.

(د) در این بخش می‌خواهیم انتخاب‌های دیگری برای شبکه استخراج ویژگی را بررسی و نتایج حاصل از شبکه‌های مختلف را با یکدیگر مقایسه و تحلیل کنیم. در این بخش برای استخراج ویژگی از شبکه VGG16 و یکبار هم از شبکه ResNet50 استفاده کنید. بخش‌های قبلی تمرین را دوباره انجام

دهید و لایه‌های مربوط به طبقه‌بندی را مجدداً آموزش دهید. تعداد پارامترها و دقت‌های نهایی حاصل از VGG16 و ResNet50 را با مقادیر حاصل از MobileNetV2 مقایسه و تحلیل کنید. (تعداد پارامترها و اینکه چه بخشی از آنها قابل یادگیری است را با استفاده از Pytorch گزارش کنید.)

(ه) در بخش‌های قبلی تمرین از وزن‌های از قبل آموزش‌دیده برای بخش استخراج ویژگی استفاده کردید. در این بخش می‌خواهیم بررسی کنیم که این دانش اولیه حاصل از Transfer Learning چقدر برای مسئله طبقه‌بندی ما مفید بوده است. برای این منظور، دوباره شبکه MobileNetV2 را در نظر بگیرید ولی این بار از وزن‌های تصادفی برای مقداردهی اولیه استفاده کنید. لایه‌های مربوط به طبقه‌بندی را اضافه کنید و شبکه را بطور کامل از ابتدا بر روی دیتاست Oxford 102 Flower آموزش دهید. مشابه با قبل، نمودار تغییرات دقت و هزینه را برای داده‌های آموزشی و ارزیابی در حین آموزش شبکه و همچنین دقت و هزینه نهایی برای داده‌های آموزشی، ارزیابی و تست را پس از تکمیل آموزش شبکه گزارش کنید. دقت‌های نهایی را با بخش ج مقایسه و تحلیل کنید.