بسمه تعالی

دانشکده مهندسی کامپیوتر

یادگیری عمیق

نام استاد: دکتر محمدی

تمرین دهم

آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

آذر ۱۴۰۰

کد های مربوط به این تمرین در نوتبوک HW10.ipynb پیوست شده است.

الف)

ابتدا کدهایی را اضافه کردم تا بتوانم وزن های هر مدلی را هنگام آموزش ذخیره کنم. بدین صورت در شرایطی که اجرا متوقف شود یا مشکل خاصی به وجود آید، میتوانیم اجرا را از همان epoch قبلی ادامه دهیم. به این صورت وزن های مدل و optimizer را در local ذخیره میکنیم:

```
if epoch % 5 == 0:
          checkpoint = {'state_dict': model.state_dict(), 'optimizer': optimiz
er.state_dict()}
          save_checkpoint(checkpoint, f'/content/checkpoint_{epoch}.pth')
```

البته روش هایی مانند mount کردن درایو هم امکان داشت که چون محیط اجرای کد را برای این تمرین تغییر دادم و از گوگل کولب استفاده نمیکردم (به علت محدودیت های GPU) این کار کمی شخت میشد و ذخیره آن ها در همان محل اجرای نوتبوک کفایت می کند.

بقیه بخش های کد همان بخش هایی هستند که از قبل به ما داده بودید، اما این خط را از True به false تغییر دادم زیرا هدف این بخش آموزش resnet با وزن های رندوم بود و نه از قبل آموزش داده شده.

```
res_mod_rand_weights = models.resnet50(pretrained=False)
```

سپس مدل را آموزش دادم و چون این روند خیلی زمانبر بود به اندازه 10 ایپاک آموزش دادم.(عدد ۹ به این خاطر نوشته شده که از ۰ شروع کردیم و نه ۱) سپس مجددا به اندازه 10 epoch آموزش دادم و بار دوم از وزن های سیو شده بار اول استفاده کردم. نتایج بعد از این ۲۰ ایپاک به این صورت بودند :

```
Epoch 8/9
----------
Iterating through data...
train Loss: 132.8857 Acc: 0.0427
Iterating through data...
val Loss: 140.9266 Acc: 0.0340

Epoch 9/9
----------
Iterating through data...
train Loss: 132.2082 Acc: 0.0429
Iterating through data...
val Loss: 140.8056 Acc: 0.0348
=> saving checkpoint
```

دقت مدل همچنان بسیار پایین است. چون بسیار مسئله بزرگی است و تعداد پارامترهای resnet50 هم حدود ۲۴ میلیون است و طبیعتا آموزش آن ها فرایندی طولانیست.

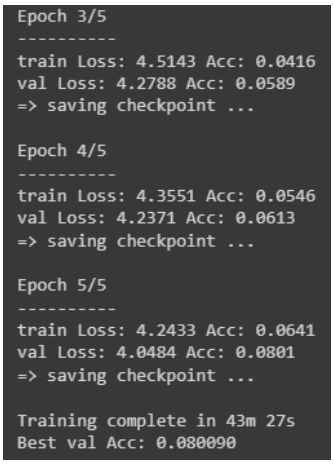
ب)

برای تابع آموزش، از تابع توسعه داده شده در سایت پایتورچ (لینک) استفاده کردم. Feature extraction به معنی صفر کردن برخی گرادیان هاست چون آن ها قبلا آموزش دیده اند. که با توابعی که تعریف کردمه ام این کار به خوبی انجام می شود. با true یا false کردن یک متغیر برای هر کدام از پارامتر ها نشان میدهیم که نیاز به محاسبه گرادیان برای آنها داریم یا خیر.

شبکه کوچکی که در انتهای لایه های resnet(pre_trained= true) قرار داده ام، دارای ۲ لایه dense با به ترتیب ۱۰۲۴ و ۱۹۶ نورون است. که اولی تابع فعالسازی Relu و دومی هم برای جدا کردن این ۱۹۶ کلاس خروجی و تخمین احتمال هریک است.

```
The Input Of Last Layer in ResNet50 has : 2048 neurons
The Output Of Last Layer in ResNet50  has : 1000 neurons

built classifier ...
all_params_WITHOUT_classifier : 161
trainable_params_BEFORE_freeze : 161
=====================================
all_params_WITH_classifier : 163
trainable_params_AFTER_freeze : 4
```

این شبکه را هم به مدت ۵ ایپاک اجرا کردم (باز هم به دلیل اجرای طولانی نشد بیشتر ران کنم) و نتیجه زیر را داشت:



پ)

ابتدا توضیحاتی درمورد svm: مدل‌های یادگیری تحت نظارت با الگوریتم‌های یادگیری مرتبط هستند که داده‌ها را برای طبقه‌بندی و تحلیل رگرسیون تجزیه و تحلیل می‌کنند. SVM ها یکی از قوی ترین روش های پیش بینی هستند که بر اساس آماری است. با توجه به مجموعه‌ای از مثال‌های آموزشی که هر کدام به عنوان متعلق به یکی از دو دسته مشخص شده‌اند، یک الگوریتم آموزشی SVM مدلی را ایجاد می‌کند که نمونه‌های جدیدی را به یک دسته یا دسته دیگر اختصاص می‌دهد و آن را به یک طبقه‌بندی‌کننده خطی باینری غیراحتمالی تبدیل می‌کند (اگرچه روش‌هایی مانند Platt مقیاس بندی برای استفاده از SVM در یک تنظیم طبقه بندی احتمالی وجود دارد). SVM نمونه های آموزشی را به نقاطی در فضا ترسیم می کند تا عرض شکاف بین دو دسته را به حداکثر برساند. نمونه‌های جدید سپس در همان فضا نگاشت می‌شوند و پیش‌بینی می‌شود که بر اساس کدام سمت شکاف به دسته‌ای تعلق دارند.

علاوه بر انجام طبقه‌بندی خطی، SVMها می‌توانند به طور موثر یک طبقه‌بندی غیرخطی را با استفاده از آنچه که ترفند هسته نامیده می‌شود، انجام دهند، و به طور ضمنی ورودی‌های خود را در فضاهای ویژگی با ابعاد بالا نگاشت می‌کنند. هنگامی که داده ها بدون برچسب هستند، یادگیری supervised

امکان پذیر نیست، و یک رویکرد یادگیری بدون نظارت مورد نیاز است، که تلاش می کند خوشه بندی طبیعی داده ها را به گروه ها بیابد و سپس داده های جدید را به این گروه های تشکیل شده ترسیم کند.

ما هم در این سوال خروجی شبکه resnet قبلی را به عنوان ورودی svm که برای توسعه آن از کتابخانه scikit-learn استفاده کرده ام. در این قسمت چون محاسبه loss امکان پذیر نبود فقط دقت را بررسی کرده ام و loss صرفا برابر صفر در نظر گرفته ام. چون svm نیازی به ضرر و محاسبه گرادیان و ... ندارد.

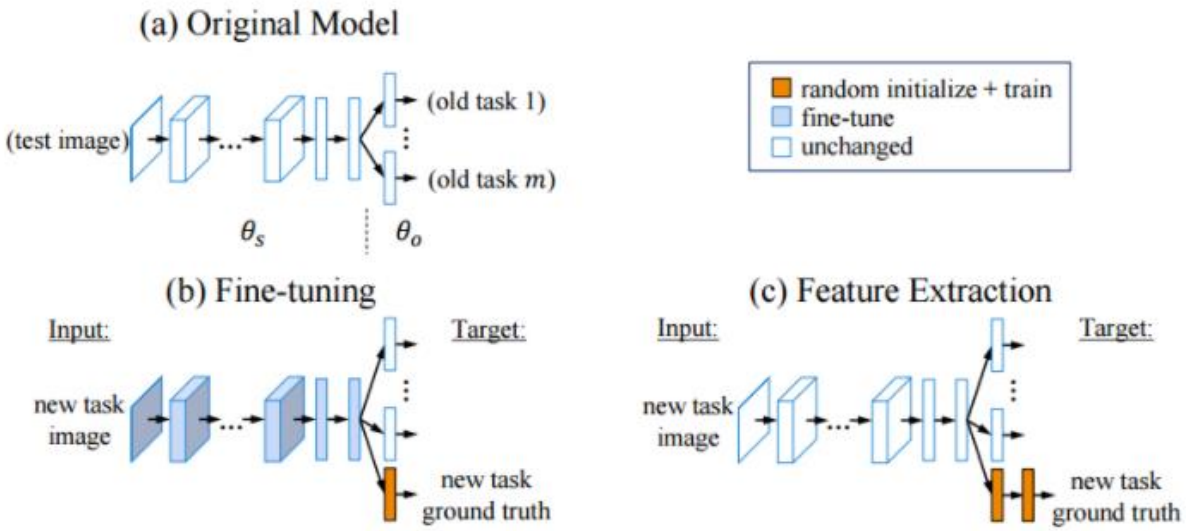
```
Epoch 7/10
----------
train Loss: 0.0000 Acc: 0.1647

Epoch 8/10
----------
train Loss: 0.0000 Acc: 0.1725

Epoch 9/10
----------
train Loss: 0.0000 Acc: 0.1490

Epoch 10/10
----------
train Loss: 0.0000 Acc: 0.1490
```

ت)

توضیح در مورد fine-tuning: تنظیم دقیق به معنای گرفتن وزن یک شبکه عصبی آموزش دیده و استفاده از آن به عنوان مقداردهی اولیه برای مدل جدیدی است که بر روی داده های همان دامنه (اغلب به عنوان مثال تصاویر) آموزش داده می شود. برای سرعت آموزش و غلبه بر اندازه مجموعه کوچک استفاده می شود.

استراتژی های مختلفی وجود دارد، مانند آموزش کل شبکه اولیه یا freeze کردن برخی از وزنه های از پیش آموزش داده شده (معمولاً لایه هایdense).

تفاوت این قسمت با feature extraction:



همانطور که در شکل زیر نشان داده شده است، در استراتژی fine tune، تمام وزن ها هنگام تمرین روی کار جدید تغییر می کنند (به جز وزنه های آخرین لایه ها برای کار اصلی)، در حالی که در استراتژی feature extraction فقط وزنه ها از آخرین لایه های اضافه شده جدید در طول مرحله آموزش تغییر می کند.

در نهایت با وجود فقط ۵ ایپاک، مطابق انتظار به نتایج بهتری در این بخش میرسیم:

```
Epoch 3/5
----------
train Loss: 3.5714 Acc: 0.1240
val Loss: 3.5490 Acc: 0.1429
=> saving checkpoint ...

Epoch 4/5
----------
train Loss: 3.2155 Acc: 0.1670
val Loss: 3.6129 Acc: 0.1434
=> saving checkpoint ...

Epoch 5/5
----------
train Loss: 3.0019 Acc: 0.2052
val Loss: 3.1011 Acc: 0.1957
=> saving checkpoint ...
```

ث)

متاسفانه پیاده سازی این قسمت را به مشکل خوردم و کدم اجرا نشد. فقط sample را انتخاب کرده ام و موفق به visualize کردن لایه های مختلف نشدم چون پایتورچ دسترسی مستقیم به خروجی لابه ها نمیداد!

منابع:

https://pytorch.org/

https://en.wikipedia.org/wiki/Support-vector_machine

https://stats.stackexchange.com/questions/331369/what-is-meant-by-fine-tuning-of-neural-network

https://stats.stackexchange.com/questions/255364/fine-tuning-vs-joint-training-vs-feature-extraction