

شبکه های عصبی
دانشگاه فردوسی مشهد

تمرین 4

نیمسال دوم تحصیلی 1402 - 1403

گروه مهندسی کامپیوتر

تیمینه توکلی 9912762267

هدف

در این تمرین قصد داریم تا یک سیستم Face Classifier و یک سیستم Face Verifier را با استفاده از وزن های از پیش آموزش داده شده مدل FaceNet طراحی کنیم.
برای سیستم تشخیص چهره از وزن های یادگیری شده مدل FaceNet بر روی دیتاست VGGFace2 استفاده میکنیم و بر روی داده های دیگری که در اختیار داریم fine tune میکنیم.
سپس با استفاده از embedding های یادگیری شده در این بخش، یک سیستم Face Verifier طراحی میکنیم.

دیتاست

دیتاست شامل تصاویر صورت مربوط به 14 هنرپیشه، با اندازه 160×160 است. برای هر فرد، حدود 100 عکس موجود است. 80 درصد از دیتاست را به داده های train و 20 درصد را به داده های test اختصاص میدهیم. اندازه هر بچ 32 در نظر گرفته شده است.

طراحی Face Classifier

در این بخش از وزنه های یادگیری شده مدل InceptionResnetV1 به عنوان یک مدل pretrained بر روی دیتاست VGGFace2 استفاده میکنیم.
بعد از ایجاد مدل و لود کردن وزن ها، در انتهای مدل، یک شبکه MLP کوچک اضافه شده که ورودی با اندازه بردار embedding ها (512) و خروجی به اندازه تعداد کلاس های دیتاست (14) دارد.

• یافتن مناسبترین تنظیم شبکه MLP

شبکه MLP را با تعداد لایه های FC متفاوت و تعداد نورون های متفاوت در هر لایه و با تکنیک های batch normalization و dropout آموزش داده و دقت شبکه را بررسی میکنیم.
شبکه نهایی:

MLP6 (batch normalization and drop out)

loss function = cross entropy

optimizer = SGD, learning rate = 0.001, momentum = 0.9

test accuracy = 100%

train accuracy = 99.05%

جدول 1- دقت train و test برای شبکه‌های مختلف

	model	bn	drop out	epochs	Test acc	Train acc
1	MLP1-tiny	False	False	5	9.28	12.58
2	MLP2-tiny	True	True	5	9.28	10.86
3	MLP3-tiny	False	False	5	6.53	8.70
4	MLP4	False	True	5	9.97	11.81
5	MLP5	True	False	5	100	99.02
6	MLP6	True	True	5	100	99.05

- تاثیر freeze کردن لایه های InceptionResnetV1 یا fine-tune کردن لایه های انتهایی آن بر زمان یادگیری

device(type='cpu')

unfreeze → 1428.956 s

fine tune last layers → 663.449 s

freeze all parameters → 540.681 s

اگر مدل InceptionResnetV1 را فریز نکنیم و هر ایپاک همه پارامترهای مدل آپدیت شود زمان یادگیری حدود 23 دقیقه است. با توجه به اینکه این مدل پارامترهای زیادی دارد، این یادگیری بسیار زمانبر است. در صورت fine tune کردن لایه‌های آخر و فریز کردن بقیه لایه ها زمان یادگیری حدود 11 دقیقه است. در صورت فریز کردن همه پارامترهای مدل InceptionResnetV1 زمان یادگیری حدود 9 دقیقه است.

طراحی Face Verifier

در این بخش، با استفاده از embedding های یادگیری شده در بخش قبلی، یک سیستم Verifier Face ایجاد میکنیم.

50 درصد از داده‌های test را به عنوان دسته تصاویر مشخص و 50 درصد را به عنوان دسته تصاویر نامشخص تقسیم‌بندی میکنیم.

بردارهای embedding را برای هر دسته محاسبه میکنیم. فاصله بردار embedding یک تصویر نامشخص را با بردار embedding تمامی تصاویر مشخص محاسبه میکنیم. اگر شباهت‌ها از یک threshold پایینتر باشند، عدم مطابقت گزارش میشود. کلاس مربوط به نزدیک‌ترین بردار تصویر مشخص را به عنوان کلاس تصویر نامشخص انتخاب میکنیم. این کلاس را میتوانیم با label واقعی تصویر مقایسه کنیم تا متوجه شویم که embedding های یاد گرفته شده توسط مدل چقدر درست هستند. با توجه به نتایج، 96.875 درصد از کلاس‌های پیش‌بینی شده برای تصاویر نامشخص، س برابر کلاس واقعی آن تصاویر هستند. با توجه به اینکه دقت مدل روی داده تست 100 درصد است، چنین نتیجه‌ای قابل انتظار بوده است.

• محاسبه فاصله بردارهای embedding با استفاده از cosine similarity یا فاصله اقلیدسی

فاصله کسینوسی (Cosine Similarity)

این معیار شباهت بین دو بردار را بر اساس زاویه‌ای که بین آن‌ها تشکیل می‌شود، محاسبه می‌کند. فاصله کسینوسی برای دو بردار a و b به صورت زیر تعریف می‌شود:

$$\text{Cosine Similarity} = a \cdot b / |a| |b| \quad (\text{formula 1})$$

که در آن $a \cdot b$ ضرب داخلی دو بردار و $|a| |b|$ نرم آن‌ها (مقدار مطلق بردار) است. مقدار فاصله کسینوسی بین -1 و 1 قرار دارد، که برای بردارهای کاملاً متفاوت -1 و برای بردارهایی که روی یکدیگر قرار دارند 1 است.

فاصله اقلیدسی (Euclidean Distance)

این معیار شباهت را بر اساس فاصله جغرافیایی بین دو نقطه در فضای چند بعدی محاسبه می‌کند. فاصله اقلیدسی بین دو بردار a و b به صورت زیر تعریف می‌شود:

$$\text{Euclidean}(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (\text{formula 2})$$

فاصله اقلیدسی برای بردارهایی که در نزدیکی یکدیگر هستند، کمتر و برای بردارهایی که از هم دور هستند، بیشتر است.

هر دو این معیارها مزایا و معایب خود را دارند و بسته به مساله مورد استفاده ممکن است انتخاب شوند. به طور کلی، فاصله کسینوسی برای مواردی مناسب است که مهم نباشد بردارها چه مقدار بزرگ یا کوچکی دارند و تنها جهت آن‌ها مهم باشد، در حالی که فاصله اقلیدسی برای مواردی مناسب است که اندازه و مقدار دقیق بردارها نیز اهمیت دارد.

فاصله کسینوسی مقاومت به تغییر اندازه دارد، به این معنی که اگر بردارهای embedding با یک ضرب عددی تغییر اندازه کنند اما جهت آن‌ها همان باقی بماند، شباهت آن‌ها تغییر نمی‌کند. این ویژگی مهم است زیرا در برخی مسائل، اندازه دقیق بردارهای embedding اهمیت کمتری دارد. همچنین بردارهای embedding معمولاً نرمال شده‌اند، به طوری که هر بردار نرم یک دارد. این باعث می‌شود فاصله کسینوسی به جای فاصله اقلیدسی مناسب تر باشد، زیرا نورمالیزه بودن بردارها باعث می‌شود تنها جهت بردارها مهم باشد و اندازه آن‌ها مهم نباشد.

بنابراین، معیار فاصله کسینوسی برای بردارهای embedding مناسب تر به نظر می‌رسد.