

دسته بندی تصاویر با الگوریتم **KNN**

مبانی برنامه نویسی

مهلت فاز اول : یازدهم آذرماه

مهلت فاز دوم : بیست و ششم آذرماه

مهلت فاز سوم: *****

دکتر یزدیان دهکردی ،دکتر زینب مهرنهاد

دستیاران: پوریا عظیمی،احمدرضا کاظمی،محسن غلامی

مقدمه

مبانی برنامه نویسی و برنامه سازی یکی از مهم ترین دوره ها برای دانشجویان مقطع کارشناسی است. نه تنها دوره های دیگر مانند برنامه نویسی پیشرفته، ساختمان داده، مبانی الگوریتم ها و بسیاری از موضوعات پر کاربرد نیازمند سرفصل های این دوره اند، بلکه برای موفقیت دانشجویان مهندسی کامپیوتر در شغل آیندشان لازم می باشد. آموزنده ترین بخش این دوره، پروژه آن است. دانشجویان باید یک برنامه کامپیوتری کامل را از پایه اجرا کنند. به خاطر داشته باشید که این برنامه فاقد رابط کاربری گرافیکی (GUI) است و معمولاً داده ای در دیتابیس (پایگاه داده) ذخیره نمی کند. در ترم های پیشین، پروژه یک برنامه کامپیوتری کلاسیک مثل برنامه کنترل فرودگاه بود که به خوبی با دوره مرتبط بود. برای نوشتن چنین برنامه هایی، دانشجویان باید بتوانند از دستورهای اساسی و فرمان های ساده ای مانند حلقه های تکرار (loops)، شرط if، تابع ها (methods)، آرایه ها (arrays)، ساختارها (structures) به طور موثر و کاربردی استفاده کنند؛ اما مشکل این بود که هیچکدام از پروژه های قبلی، به زندگی واقعی نزدیک نبودند. به همین دلیل، این ترم ما توجهمان را به پروژه ای کاربردی تر دادیم.

یادگیری ماشین چیست؟

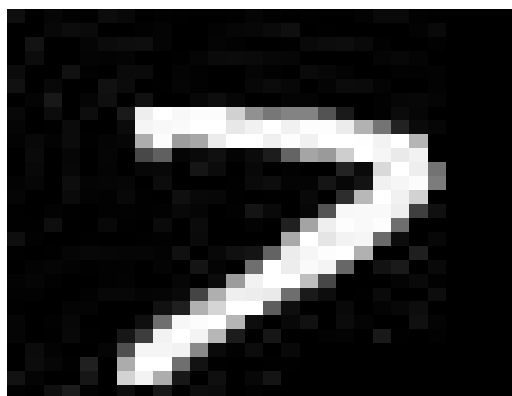
امروزه هوش مصنوعی (AI) به یکی از کاربردی ترین رشته های کامپیوتر تبدیل شده است. بسیاری از صنایع، برای استفاده از هوش مصنوعی در تلاشند تا عملکرد بهتری داشته باشند. یادگیری ماشین یک زیرمجموعه هوش مصنوعی است که سعی می کند الگوهای مفید و کاربردی را از داده ها بیاموزد. به عبارتی دیگر، ما صراحتاً به الگوریتم نمی گوییم که چگونه کار خاصی را انجام دهد بلکه الگوریتم باید از داده ها، الگو را یاد بگیرد تا وظیفه اش را به انجام برساند.

به عنوان مثال، مشکل تایید و شناسایی هویت از یک تصویر ورودی را در نظر بگیرید. الگوریتم باید تشخیص دهد دو تصویر ورودی، متعلق به یک شخص هستند یا نه. در این مورد، ما تنها به الگوریتم، چند تصویر از یک شخص معین و یا چند تصویر از اشخاص مختلف می دهیم و الگوریتم باید بتواند تشخیص دهد کدام تصاویر متعلق به همان شخص هستند و کدام یک نیستند. به طوری که اگر به الگوریتم تعدادی تصویر دادیم که قبلاً توسط او دیده نشده است، توانایی تشخیص اینکه عکس ها به همان شخص تعلق دارند یا نه را داشته باشد.

دسته بندی تصاویر

دسته بندی تصویر، کار طبقه بندی یک تصویر ورودی، میان دسته های مشخص است. با دادن یک تصویر ورودی، ما باید یک برچسب از مجموعه برچسب های از پیش تعریف شده، روی تصویر قرار دهیم. برای مثال، دسته بندی یک عکس (بر پایه اینکه کدام عدد در آن وجود دارد) را در نظر بگیرید. به طور دقیق تر، به الگوریتم، یک تصویر به عنوان ورودی داده می شود و الگوریتم باید پیش بینی کند کدام عدد از بین ده عدد مختلف صفر، یک، دو و... در تصویر وجود دارد.

به عنوان مثال، با دادن ورودی زیر، الگوریتم باید پیش بینی کند که یک عدد 7 در عکس وجود دارد:



نمای کلی پروژه

در این پروژه، شما باید یک «طبقه بندی کننده تصویر» پیاده سازی کنید که روی هر دو دیتاست **mnist** (مربوط به تشخیص رقم) و **fashion mnist** (مربوط به تشخیص لباس) کار می کند. به طور دقیق تر، برنامه شما باید بتواند یک تصویر ورودی را دریافت و یک کلاس (**class**) به آن اختصاص دهد و در کلاس مربوط به آن تصویر قرار دهید. (برای دسته بندی تصاویر باید از الگوریتم **KNN** استفاده کنید.) این پروژه در سه فاز مختلف اجرا خواهد شد. در هر فاز، شما باید بخش هایی از پروژه را به اجرا درآورید. در پایان فاز سوم، برنامه شما تکمیل خواهد شد. لازم به ذکر است که پروژه نه تنها تاثیر زیادی روی نمره نهایی شما دارد، بلکه به شما در یادگیری عمیق تر و بهتر مفاهیم برنامه نویسی کمک می کند.

نیازمندی ها

برنامه نهایی شما باید قادر باشد کارهای زیر را انجام دهد:

تصویر ورودی را دسته بندی کند (تصویر ورودی ممکن است متعلق به دیتاست **mnist** یا **mnist fashion** باشد).

تنظیم پارامترهای الگوریتم - پارامترهای **K**

ویژگی ها را از تصویر ورودی و ترینینگ ست ها استخراج کند.

ویژگی های استخراج شده از ترینینگ ست ها را در یک فایل ذخیره کند.

ویژگی های استخراج شده از فایل را بخواند

همچنین، توجه داشته باشید که طبقه بندی تصویر ورودی نباید بیشتر از ده ثانیه طول بکشد. (هنگام استخراج ویژگی های عکس)

فاز اول

در این فاز، شما فقط باید منوهای برنامه خود را پیاده سازی کنید. برای این مرحله منوهای زیر را خواهید نوشت:

- ← **منوی شروع:** هنگامی که برنامه شما شروع به کار کرد، این منو باز خواهد شد.
- ← **منوی اصلی (Main menu):** این منو، دارای تعدادی منوی فرعی است که هر کدام کار مشخصی انجام می دهند:

- **جستجو و کاوش تصاویر را از دیتاست (مجموعه داده):** کاربر برنامه شما باید بتواند دیتاست، ترین، تست ست، کلاس تصویر و عدد تصویر را انتخاب کند.
- **ترین (Train):** ترین در این منو، کاربر می تواند استخراج ویژگی ها برای ترینینگ ست را شروع و ویژگی های استخراج شده را در یک فایل جدید یا فایل بارگذاری شده ذخیره کند. همچنین، در این منو، کاربر باید بتواند پارامتر k را تنظیم کند. (می توانید منوهای فرعی را به طور دلخواه طراحی کنید).
- **تست:** در این منو کاربر باید بتواند یک دیتاست را انتخاب کند و پس از وار کردن آدرس تصویر ورودی، نتایج طبقه بندی را ببیند.
- **خروج:** این گزینه، برنامه را به اولین منو باز می گرداند. (منوی شروع)

← **اطلاعات برنامه نویسی (info):** زمانی که این منو باز می شود، اطلاعات شما (نام، نام خانوادگی، ایمیل) باید بر روی صفحه نمایش چاپ شده باشد.

← **خروج (Exit):** این گزینه برنامه را کاملاً می بندد¹.

¹ prototype.exe را برای جستجو در منو ها باز کنید. (شما باید چیزی شبیه به این فایل را برای این مرحله پیاده سازی کنید)

فاز دوم

در این فاز، بخش هایی از الگوریتم واقعی **KNN** را پیاده سازی خواهید کرد.
قبل از شروع بحث ما در مورد این فاز، بیایید به کارکرد کلی الگوریتم **KNN** نگاهی بیندازیم:

الگوریتم KNN

KNN که مخفف **K-nearest neighbors** (نزدیک ترین همسایگان **k**) است، یک الگوریتم دسته بندی است که نمونه های ورودی را کلاس بندی و طبقه بندی می کند. در این مسئله، ما یک تصویر ورودی را بر این اساس طبقه بندی میکنیم که کلاس (**class**) نزدیک ترین همسایگان این داده با توجه به داده های پیشین (یعنی همان دیتاست) کدام است .

به بیان دقیق تر، در نظر بگیرید که به شما **1000** آرایه از اعداد داده شده است که هر یک از آنها **10** عنصر دارند و به هر یک از آنها یک کلاس اختصاص داده شده است. با استفاده از این **1000** آرایه، ما می خواهیم با بهره گیری از این نقاط داده (**datapoint**)، آرایه ورودی دیگری را دسته بندی کنیم. برای انجام این کار، ما ابتدا باید فاصله هر یک از نقاط داده (**datapoint**) تا هر یک از آن **1000** نمونه را محاسبه کنیم؛ سپس کمترین فاصله های **k** را بیابیم.

به عنوان مثال، در نظر بگیرید که **5** تا از حداقل فاصله ها، برای آرایه های **1**، **2**، **20**، **50** و **100** است. آنها به ترتیب به کلاس های **1**، **1**، **2** و **1** تعلق دارند. بنابراین، نزدیکترین نقاط، متعلق به کلاس **1** خواهد بود. بنابراین، آرایه ورودی را به کلاس یک اختصاص می دهیم.

آخرین چیزی که باید مورد بحث قرار گیرد این است که چگونه فاصله بین دو آرایه را اندازه گیری کنیم. راه های زیادی برای انجام این کار وجود دارد، اما رایج ترین آنها «فاصله اقلیدسی»، «شباهت کسینوسی» و «فاصله منهتن» است. برای ادامه بحث، دو آرایه **A1** با عناصر **a1**، **a2**،...، **a10** و آرایه **B2** با عناصر **b1**، **b2**،...، **b10** را در نظر بگیرید. اولین و یکی از رایج ترین معیارهای اندازه گیری فاصله اقلیدسی است که به صورت زیر محاسبه می شود:

$$D(A, B) = \sum_{i=1}^{10} (a_i - b_i)^2$$

روش دوم به شرح زیر است (شباهت کسینوسی):

$$D(A, B) = \frac{A \cdot B}{||A|| \times ||B||}$$

به صورتی که $A \cdot B$ حاصلضرب A و B است و $||A||$ نُرم $L2$ (Norm) A است. این دو عبارت اینگونه محاسبه میشوند:

$$A \cdot B = \sum_{i=1}^{10} a_i \times b_i$$

$$||A|| = \sqrt{\sum_{i=1}^{10} a_i^2}$$

و روش آخر روش "Manhattan distance" (فاصله منهتن) به صورت زیر است:

$$D(A, B) = \sum_{i=1}^{10} |a_i - b_i|$$

در این مسئله ما از روش "Manhattan distance" (فاصله منهتن) استفاده میکنیم.

چه باید بکنیم

در این فاز شما باید موارد زیر را انجام دهید:

🚧 پیاده سازی تابعی که با گرفتن دو آرایه به عنوان ورودی ، فاصله بین این دو آرایه را حساب کند .(با استفاده از فاصله اقلیدسی)

🚧 پیاده سازی تابعی که با گرفتن ماتریس و دو آرایه ، فاصله آرایه با هر ردیف ماتریس را حساب میکند.

تابعی پیاده سازی کنید که آرایه از مقادیر **float** را با سایز **28** در **28** دریافت کرده سپس آن را به چهار شبکه(قسمت) تقسیم کند. (به صورتی که هر شبکه شامل **7*7** متغیر **float** باشد)
سپس بردار ویژگی (**feature vector of this matrix**) این ماتریس را محاسبه کنید.(برای هر خانه ی شبکه ، **Mean** و **std** را محاسبه کرده و همه ی **32** عدد را در آرایه قرار دهید).

$$Mean(A) = \frac{1}{n} \sum_{i=1}^{10} a_i$$

$$Std(A) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (a_i - Mean(A))^2}$$

- به منظور گرفتن و ذخیره عکس در ماتریس ، از تابع **load_image** که در فایل **utils.cpp** قرار دارد استفاده کنید.
- با استفاده از تابع **load_image** و تابع های دیگر از قسمت های قبل ، فاصله بین دو عکس موجود در دیتابیس را محاسبه کنید.

Buenos (بوئنوس) -----< bonus خودمون

🚀 پاک کردن صفحه هنگامی که منو باز میشود. (این مورد هم در ویندوز هم و هم در لینوکس کار کند).

🚀 استفاده از چندین فایل برای نوشتن کد خود .

🚀 استفاده از توابع تست برای محاسبات ریاضی .

🚀 ترجمه این داکيومنت به زبان فارسی و درست کردن یک **pull request** در گیت هاب.

🚀 پیاده سازی استراتژی های استخراج اطلاعات و ویژگی های بیشتر .

اصطلاحات یادگیری ماشین

Data Set (مجموعه داده): دیتا ست مجموعه ای از داده هاست که به الگوریتم آموزشی ماشین داده میشود تا با استفاده از این داده ها الگوریتم هایش را آموزش دهد. این داده ها نمونه هایی از داده های واقعی هستند که مدل باید روی آن ها کار کند.

Training Set (مجموعه آموزشی): بخشی از کل مجموعه داده است که برای آموزش الگوریتم استفاده میشود. معمولاً برای برنامه های کاربردی جهانی حدود **60** درصد کل داده برای آموزش الگوریتم استفاده میشود.

Test Set (مجموعه آزمایشی): بخشی از کل مجموعه داده است که برای تست الگوریتم استفاده میشود. معمولاً برای برنامه های کاربردی جهانی حدود **20** درصد کل داده برای تست الگوریتم استفاده میشود.

لازم به ذکر است مجموعه ای با اسم مجموعه اعتبار سنجی یادگیری ماشین (**validation set**) هم وجود دارد که فراتر از پروژه ماست.

انواع الگوریتم ها

معمولا الگوریتم های یادگیری به سه دسته اصلی نظارتی ، غیر نظارتی و تقویتی دسته بندی میشوند.

- **الگوریتم های نظارتی (Supervised algorithms)** الگوریتم هایی هستند

که علاوه بر داده ، به آنها برچسب های داده هم داده میشود . برای مثال ، در عملیات دسته بندی عکس به شما عکس ها به همراه برچسب هایشان داده میشود و الگوریتم همزمان هم دیتا و هم برچسب را برای درک الگوی مجموعه داده مورد محاسبه قرار میدهد.

اگر برچسب داده شده ، یک برچسب ادامه دار باشد، مشکل را ارور رگرسیون (regression) مینامیم. در غیر این صورت (اگر برچسب ها مجزا باشند) الگوریتم را یک مشکل طبقه بندی (classification) مینامیم.

- **الگوریتم های غیر نظارتی (Unsupervised algorithms)** نوعی از

الگوریتم است که شما فقط داده در اختیار دارید و دیگر خبری از برچسب ها نیست. الگوریتم باید الگو را تنها از طریق دیتا یاد بگیرد.

برخی از وظایف یادگیری غیر نظارتی عبارتند از یافتن خوشه ها در داده ها ، تخمین چگالی ، تشخیص ناهنجاری ، کاهش ابعاد و غیره.

- **الگوریتم های تقویتی (Reinforcement algorithms)** الگوریتم های

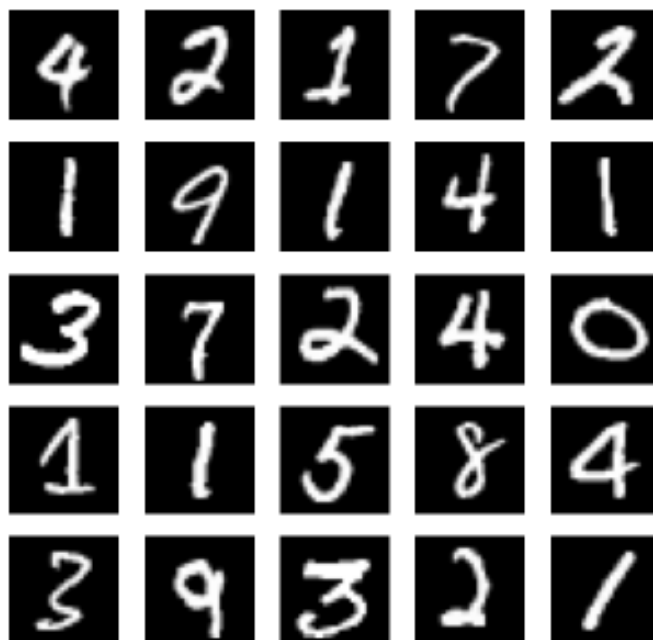
تقویت کننده الگوریتم هایی هستند که هیچ نوع داده ای ندارند ، بلکه هنگام انجام عملیات در محیط ، به آنها سیگنال پاداش داده میشود.

موقعیتی را تصور کنید که میخواهیم به یک ربات راه رفتن یاد بدهیم. در این حالت ، ما هیچ اطلاعاتی در اختیار نداریم. در این حالت بدین گونه عمل میکنیم که در شروع به سادگی از ربات میخواهیم که راه برود ، هنگامی که شروع به حرکت رو به جلو میکند به او امتیاز مثبت مثلا "1+" میدهیم. و هنگامی که حرکت اشتباهی انجام داد یا زمین خورد به او امتیاز منفی مثلا "1-" میدهیم. بعد از مدتی ، ربات برای اینکه فقط امتیاز مثبت بگیرد ، یاد میگیرد چگونه راه برود.

مجموعه داده MNIST

مجموعه داده MNIST دیتابیزی متشکل از عکس های خاکستری 28*28 پیکسلی است. در این عکس ها عددی (از 0 تا 9) نوشته شده است .

در مجموع 60,000 عکس برای آموزش الگوریتم و 10,000 عکس برای تست الگوریتم است .



مجموعه داده Fashion MNIST

مجموعه داده "Fashion MNIST" دیتاییسی متشکل از عکس های خاکستری 28×28 پیکسلی است. هر تصویر شامل عکسی از لباس از بین ده نوع مختلف لباس است. (کفش بوت ، بگی (نوعی لباس)، کت ، پولیور ، پیراهن ، تی شرت ، صندل ، کفش ورزشی) در مجموع 60,000 عکس برای آموزش الگوریتم و 10,000 عکس برای تست الگوریتم است .



موفق باشید...

برای اطلاعات بیشتر در مورد دیتابیس ها میتوانید به

<https://colab.research.google.com/drive/1u8KIFWUxx>

مراجعه کنید

[SCBtnefj61L_t4H0AP2U_K8?usp=sharing](https://colab.research.google.com/drive/1u8KIFWUxxSCBtnefj61L_t4H0AP2U_K8?usp=sharing)

❖ نکته شديدا مهم: بعد از اينكه داکيومنت رو خونديد يه پيام به پوريا اعظمی با آیدی

(@pooriaazami) بدید و "Done" رو بفرستید.

ترجمه شده توسط شایان چالشی && عرفان ابویی