

به نام خداوند بخشنده مهربان



عنوان تمرین

بردار پشتیبان SVM

عنوان درس

یادگیری ماشین

استاد

دکتر الهام قصرالدشتی

دستیاران آموزشی

مهرداد قصابی

مریم صفوی

گردآورنده

سید حسین حسینی

بهار ۱۴۰۴

دانشکده مهندسی کامپیوتر

دانشگاه اصفهان

## مستندسازی پروژه: طبقه‌بندی ارقام دست‌نویس (MNIST) با ماشین بردار پشتیبان (SVM)

### مقدمه

این پروژه با هدف ساخت یک مدل یادگیری ماشین برای تشخیص ارقام دست‌نویس از مجموعه داده معروف **MNIST** طراحی شده است. مجموعه داده **MNIST** شامل ۷۰,۰۰۰ تصویر از ارقام دست‌نویس (۰ تا ۹) است که به طور گسترده به عنوان یک معیار استاندارد برای ارزیابی الگوریتم‌های طبقه‌بندی تصویر استفاده می‌شود. در این پروژه، از الگوریتم **ماشین بردار پشتیبان (Support Vector Machine - SVM)** برای طبقه‌بندی این تصاویر استفاده شده است. **SVM** یک الگوریتم یادگیری نظارت‌شده قدرتمند است که با یافتن یک **اِبر صفحه (Hyperplane)** بهینه، داده‌های کلاس‌های مختلف را از یکدیگر جدا می‌کند. اِبر صفحه به گونه‌ای انتخاب می‌شود که بیشترین حاشیه (**Margin**) را بین نزدیک‌ترین نقاط داده‌ای هر کلاس (که به آن‌ها **بردارهای پشتیبان** گفته می‌شود) ایجاد کند.

پیاده‌سازی **SVM** در این پروژه از پایه و با استفاده از کتابخانه **cvxopt** برای حل مسائل بهینه‌سازی درجه دوم (**Quadratic Programming**) انجام شده است. این رویکرد به جای استفاده از کتابخانه‌های آماده مانند **Scikit-learn**، درک عمیق‌تری از مبانی ریاضی و فرآیند بهینه‌سازی در **SVM** فراهم می‌کند.

### مراحل اجرای پروژه

پروژه شامل مراحل مختلفی است که در ادامه به تفصیل توضیح داده می‌شوند.

#### ۱. فراخوانی کتابخانه‌ها (Libraries)

ابتدا کتابخانه‌های مورد نیاز برای پردازش داده، بهینه‌سازی و مصورسازی فراخوانی می‌شوند:

- **Numpy**: برای انجام عملیات عددی و کار با آرایه‌ها، به ویژه برای پردازش تصاویر و محاسبات ماتریسی.
- **keras.datasets.mnist**: برای بارگذاری مستقیم مجموعه داده **MNIST**.
- **matplotlib.pyplot**: برای رسم نمودارها، به خصوص ماتریس درهم‌ریختگی.
- **Cvxopt**: یک کتابخانه قدرتمند برای حل مسائل بهینه‌سازی محدب، که در این پروژه برای حل مسئله بهینه‌سازی درجه دوم (**QP**) در الگوریتم **SVM** استفاده می‌شود.
- **sklearn.metrics**: برای ایجاد و نمایش ماتریس درهم‌ریختگی جهت ارزیابی مدل.

## ۲. بارگذاری و آماده‌سازی داده‌ها (Load and Prepare Data)

1. بارگذاری داده‌های MNIST : مجموعه داده MNIST به صورت خودکار از طریق keras دانلود و به دو بخش آموزش ( $X_{train}, y_{train}$ ) و آزمون ( $X_{test}, y_{test}$ ) تقسیم می‌شود.
2. مسطح‌سازی تصاویر (Flattening) : هر تصویر در مجموعه داده MNIST ابعادی برابر  $28 \times 28$  پیکسل دارد. برای اینکه بتوان این تصاویر را به عنوان ورودی به الگوریتم SVM داد، هر تصویر به یک بردار تک‌بعدی با  $28 \times 28 = 784$  ویژگی تبدیل می‌شود.
3. نرمال‌سازی (Normalization) : مقادیر پیکسل‌های تصاویر بین ۰ تا ۲۵۵ هستند. برای بهبود عملکرد الگوریتم بهینه‌سازی و جلوگیری از مشکلات عددی، این مقادیر با تقسیم بر ۲۵۵ به بازه  $[0, 1]$  نرمال‌سازی می‌شوند.

## ۳. تعریف توابع اصلی SVM

در این بخش، توابع اصلی برای آموزش و پیش‌بینی با SVM از پایه پیاده‌سازی می‌شوند.

### ۳.۱. تابع کرنل خطی (Linear Kernel)

کرنل خطی ساده‌ترین نوع کرنل در SVM است که حاصل ضرب داخلی دو بردار ورودی را محاسبه می‌کند. این کرنل زمانی مناسب است که داده‌ها به صورت خطی قابل تفکیک باشند.

### ۳.۲. تابع آموزش SVM برای طبقه‌بندی دوتایی (svm\_train)

این تابع، قلب الگوریتم SVM را پیاده‌سازی می‌کند. از آنجا که SVM به طور ذاتی یک طبقه‌بند دوتایی است، این تابع برای جدا کردن دو کلاس (مثلاً یک رقم خاص در مقابل بقیه ارقام) طراحی شده است.

#### 1. فرموله‌بندی مسئله بهینه‌سازی

(Quadratic Programming) تبدیل می‌شود. این تابع، ماتریس‌ها و بردارهای مورد نیاز برای

حل‌کننده cvxopt را می‌سازد:

- $P, q$ : ماتریس‌ها و بردارهای تابع هدف.
- $G, h$ : ماتریس‌ها و بردارهای مربوط به قیود نابرابری.
- $A, b$ : ماتریس‌ها و بردارهای مربوط به قیود برابری.

2. حل مسئله QP: با استفاده از solvers.qp از کتابخانه cvxopt، ضرایب لاگرانژ (alphas) به دست می آیند.

3. یافتن بردارهای پشتیبان: نقاط داده ای که ضریب لاگرانژ آن ها بزرگ تر از یک آستانه کوچک (مثلاً  $1e-5$ ) است، به عنوان بردارهای پشتیبان (Support Vectors) شناسایی می شوند.

4. محاسبه  $w$  و  $b$ : با استفاده از بردارهای پشتیبان و ضرایب لاگرانژ آن ها، پارامترهای ابرصفحه، یعنی بردار وزن ( $w$ ) و بایاس ( $b$ )، محاسبه می شوند.

### ۳.۳. تابع پیش بینی (svm\_predict)

این تابع با استفاده از پارامترهای  $w$  و  $b$  که از مرحله آموزش به دست آمده اند، برای یک ورودی جدید ( $X$ ) پیش بینی انجام می دهد. خروجی این تابع، فاصله نقطه از ابرصفحه است. علامت این خروجی، کلاس پیش بینی شده را تعیین می کند.

### ۴. آموزش مدل چند کلاسه با استراتژی یک در مقابل بقیه (One-vs-Rest)

از آنجا که SVM یک طبقه بند دوتایی است، برای حل مسئله طبقه بندی ۱۰ کلاسه (ارقام ۰ تا ۹) از استراتژی یک در مقابل بقیه (One-vs-Rest) استفاده می شود.

• **فرآیند کار:** برای هر رقم (از ۰ تا ۹)، یک طبقه بند SVM جداگانه آموزش داده می شود. هر طبقه بند یاد می گیرد که یک رقم خاص را از ۹ رقم دیگر تشخیص دهد. به این ترتیب، ۱۰ طبقه بند دوتایی ساخته می شود.

• **آموزش دسته ای (Batch Training):** به دلیل حجم بالای داده های آموزشی (۶۰,۰۰۰ تصویر) و محدودیت های حافظه و محاسباتی در هنگام حل مسئله QP، فرآیند آموزش برای هر طبقه بند به صورت دسته ای (batch) انجام می شود. داده های آموزشی به دسته های ۱۰۰۰ تایی تقسیم شده و مدل برای هر دسته به صورت جداگانه آموزش می بیند. در نهایت، پارامترهای  $w$  و  $b$  آخرین دسته آموزش دیده به عنوان پارامترهای نهایی آن طبقه بند در نظر گرفته می شوند.

## ۵. پیش‌بینی و ارزیابی مدل

1. **پیش‌بینی روی داده‌های آزمون:** برای هر تصویر در مجموعه آزمون، هر یک از ۱۰ طبقه‌بند یک

امتیاز (فاصله از ابرصفحه) تولید می‌کند. دسته‌ای که بالاترین امتیاز را از طبقه‌بند مربوط به خود دریافت

کند، به عنوان برچسب نهایی پیش‌بینی می‌شود.

2. **محاسبه دقت: (Accuracy)** با مقایسه برچسب‌های پیش‌بینی شده با برچسب‌های واقعی، دقت کلی مدل

محاسبه می‌شود. دقت به دست آمده در این پروژه حدود ۷۸.۶۸٪ است. این دقت ممکن است در مقایسه

با مدل‌های پیشرفته شبکه‌های عصبی کمتر باشد، اما برای یک پیاده‌سازی پایه‌ای SVM با کرنل خطی و

آموزش دسته‌ای ساده، نتیجه قابل قبولی است.

3. **ماتریس درهم‌ریختگی (Confusion Matrix):** این ماتریس به صورت بصری نشان می‌دهد که

مدل در تشخیص کدام ارقام موفق‌تر بوده و کدام‌ها را بیشتر با هم اشتباه گرفته است. به عنوان مثال، از روی

ماتریس می‌توان دید که آیا مدل تمایل دارد رقم "۴" را با "۹" یا "۳" را با "۵" اشتباه بگیرد.

### جمع‌بندی و نتیجه‌گیری

این پروژه با موفقیت یک طبقه‌بند SVM با کرنل خطی را از پایه و با استفاده از حل‌کننده cvxopt پیاده‌سازی کرد.

استراتژی **یک در مقابل بقیه** برای تبدیل مسئله به یک طبقه‌بندی چند کلاسه به کار گرفته شد و برای مدیریت حجم

بالای داده، از آموزش دسته‌ای استفاده گردید.

هرچند دقت نهایی (۷۸.۶۸٪) به اندازه مدل‌های پیشرفته نیست، اما این پروژه به خوبی مفاهیم بنیادی SVM،

چالش‌های بهینه‌سازی در مقیاس بزرگ و روش‌های مدیریت این چالش‌ها را به نمایش می‌گذارد.

برای بهبودهای احتمالی می‌توان موارد زیر را در نظر گرفت:

- استفاده از کرنل‌های غیرخطی مانند **کرنل چند جمله‌ای (Polynomial)** یا **کرنل گوسی**

- **(RBF)** که می‌توانند مرزهای تصمیم پیچیده‌تری را یاد بگیرند.

- بهبود استراتژی آموزش دسته‌ای، مثلاً با میانگین‌گیری از پارامترهای  $w$  و  $b$  هر دسته به جای استفاده از

پارامترهای آخرین دسته.

- تنظیم هایپرپارامتر  $C$  (پارامتر جریمه) برای یافتن بهترین تعادل بین بیش‌برازش و کم‌برازش.

## منابع و مراجع (References)

1. Cortes, C., & Vapnik, V. (1995). *Support-Vector Networks. Machine Learning*, 20(3), 273-297.
  - توضیح: مقاله اصلی و کلاسیک که الگوریتم ماشین بردار پشتیبان را معرفی کرد. این مقاله برای درک عمیق مبانی نظری SVM ضروری است.
2. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
  - توضیح: این کتاب یکی از مراجع اصلی در یادگیری ماشین است و فصل ۷ آن به طور کامل به ماشین‌های بردار پشتیبان، فرمولاسیون ریاضی و تکنیک‌های کرنل می‌پردازد.
3. MNIST (Modified National Institute of Standards and Technology) dataset.
  - توضیح: مجموعه داده ارقام دست‌نویس که توسط Yann LeCun و همکارانش گردآوری شده و به عنوان معیار استاندارد در حوزه بینایی ماشین و یادگیری عمیق شناخته می‌شود.
  - لینک: <http://yann.lecun.com/exdb/mnist/>
4. CVXOPT Documentation.
  - توضیح: مستندات رسمی کتابخانه cvxopt که راهنمای کاملی برای استفاده از حل‌کننده‌های بهینه‌سازی محدب، به ویژه solvers.qp، ارائه می‌دهد.
  - لینک: <https://cvxopt.org/userguide/index.html>

😊 موفق باشید 😊