

رسالة محمد

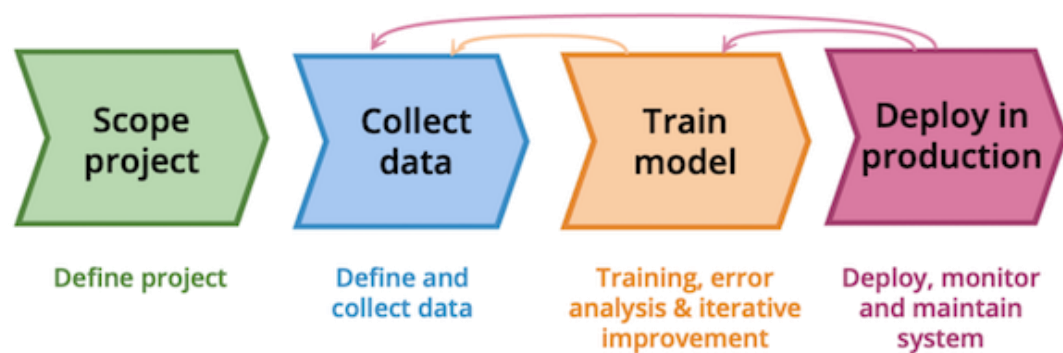
روش‌شناسی کاربردی

Practical Methodology

روش‌شناسی کاربردی

- بکارگیری موفقیت‌آمیز تکنیک‌های یادگیری عمیق به چیزی بیش از دانش کافی از الگوریتم‌ها و نحوه کار آنها نیاز دارد
- چگونه یک الگوریتم را برای یک کاربرد خاص انتخاب کنیم و چگونه می‌توان به بازخورد بدست آمده از آزمایش‌ها نظارت کرد و به آنها پاسخ داد؟

Lifecycle of an ML Project



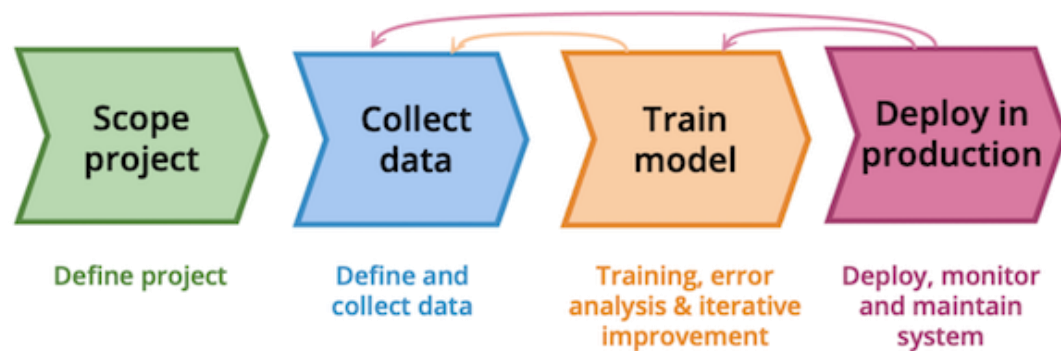
- تصمیم برای جمع‌آوری داده‌های بیشتر
- افزایش یا کاهش ظرفیت مدل
- افزودن یا حذف منظم‌سازی پارامترها
- بهبود بهینه‌سازی یک مدل
- اشکال‌زدایی نرم‌افزاری

فرآیند طراحی



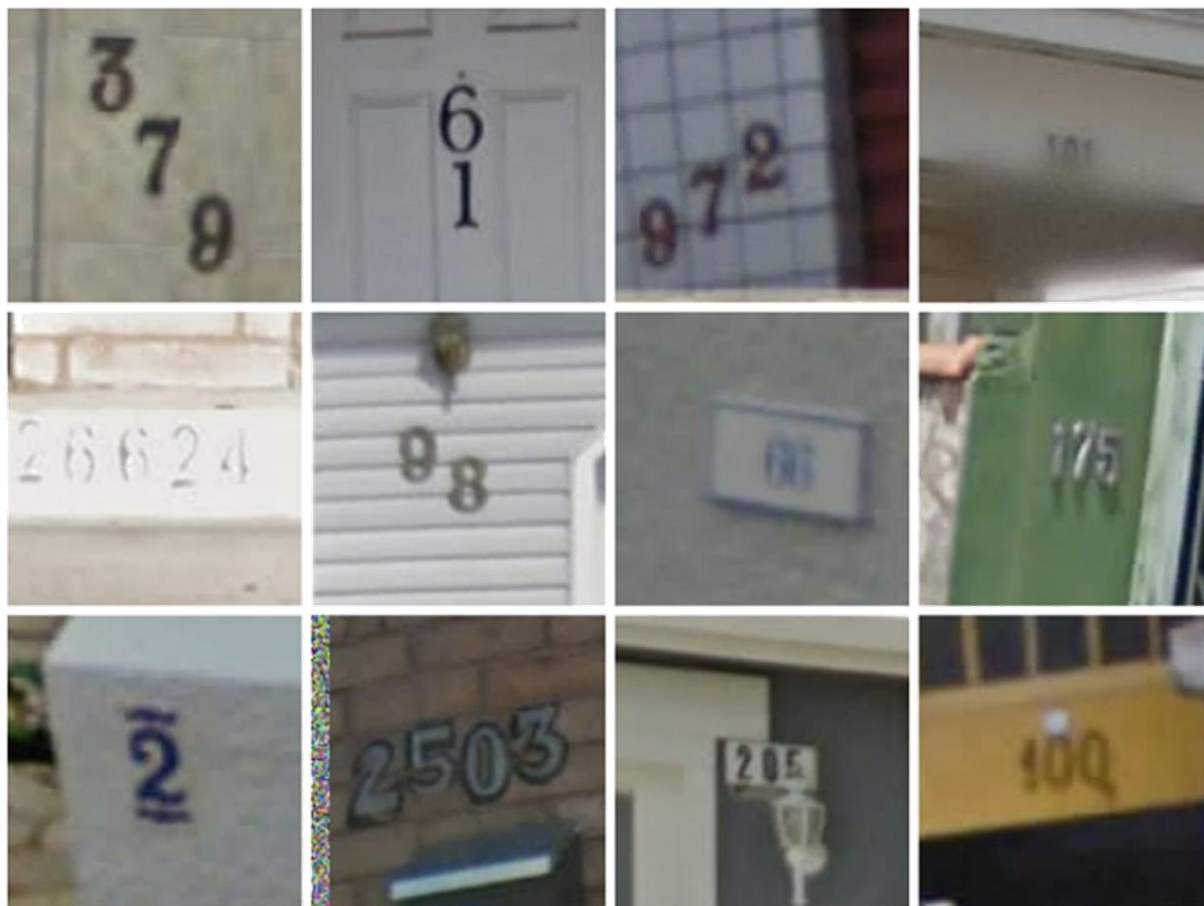
- اهداف خود را مشخص کنید
- در اسرع وقت یک مدل پایه end-to-end ایجاد کنید
- تشخیص دهید کدام بخش‌ها ضعیف‌تر از حد انتظار عمل می‌کنند
- به طور مکرر تغییرات تدریجی ایجاد کنید

Lifecycle of an ML Project

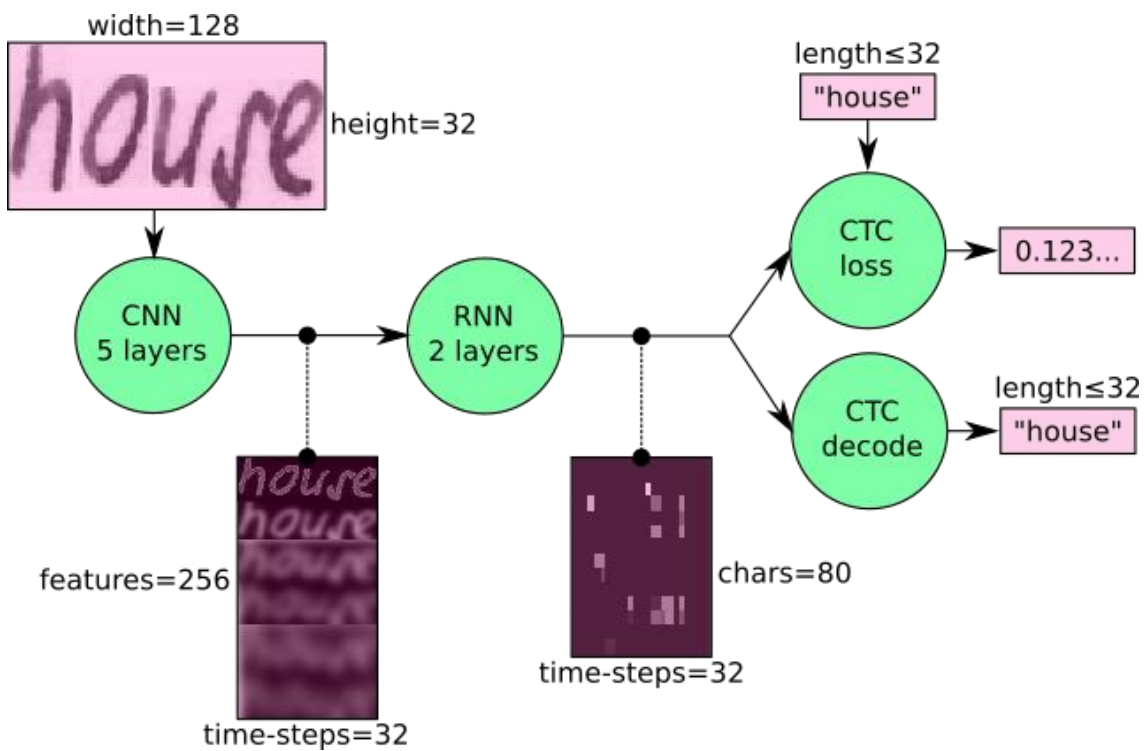




مثال: خواندن شماره پلاک ساختمان‌ها

- هدف افزودن پلاک ساختمان‌ها به نقشه گوگل است
- خودروهای Street View از ساختمان‌ها تصویر می‌گیرند و مختصات GPS مرتبط با هر تصویر را ثبت می‌کنند



OCR

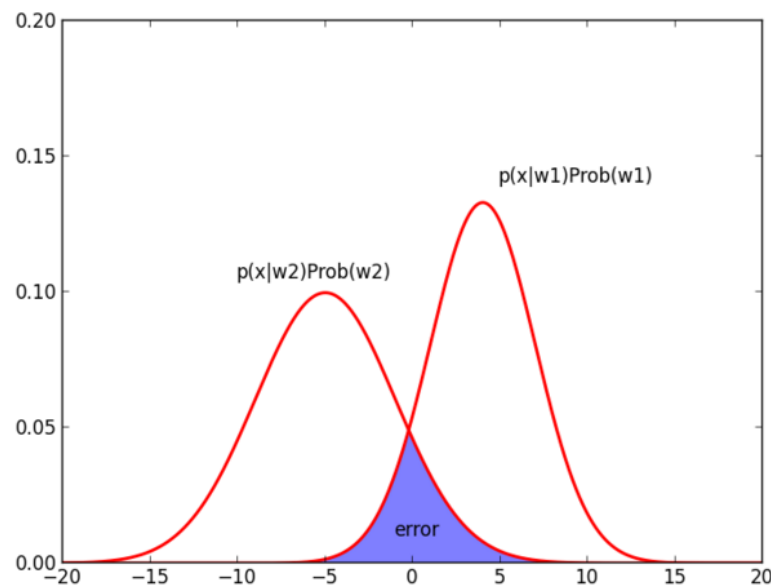


تصویر مرجع	ناحیه برش خورده (ورودی مدل)	خروجی مطلوب
		۶۱۰۴۳۳۷۰۱۸۵۶۰۵۷۱
		۱۲۳۴۵۶۷۸۹۰
		۰۰۶۷۰۶۲۵۷۱
		۶۰۳۷۹۹۱۱۶۵۳۱۶۲۹۴

معیارهای ارزیابی عملکرد

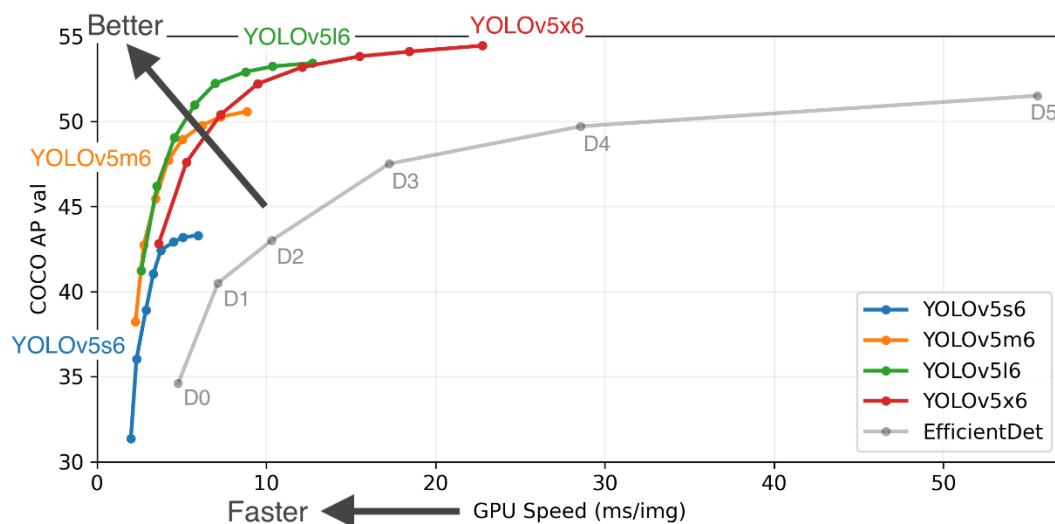


- اهداف خود را مشخص کنید
 - از چه معیاری استفاده شود؟
 - چه سطحی از عملکرد مورد نظر است؟
- همه اقدامات آینده خود را با معیار خطا هدایت کنید
- هیچ برنامه‌ای به خطای صفر دست پیدا نمی‌کند!
- حجم داده‌های آموزشی به دلایل مختلفی محدودیت دارد
 - زمان، پول، دشواری



معیارهای ارزیابی عملکرد

- چگونه می‌توان سطح معقولی از عملکرد مورد انتظار را تعیین کرد؟



<https://github.com/ultralytics/yolov5>

- در محیط دانشگاهی

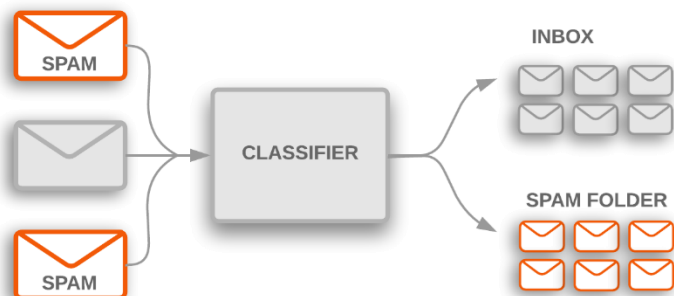
- نرخ خطای قابل دستیابی بر اساس نتایج منتشر شده

- در محیط صنعتی

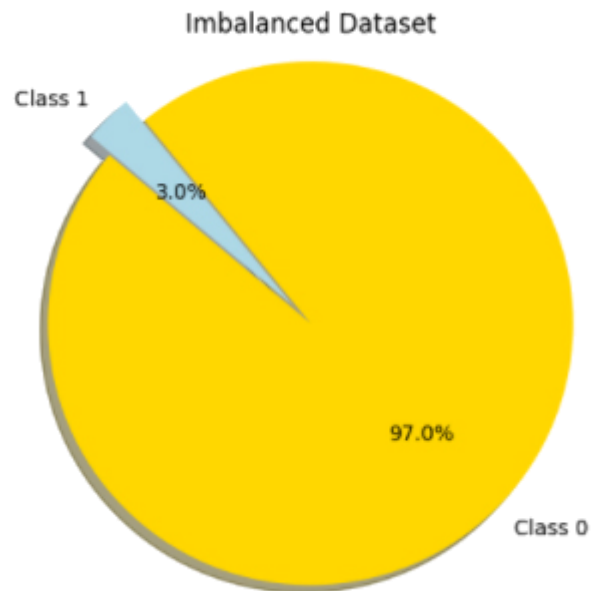
- در مورد میزان حداکثر خطای ممکن برای ایمن، مقرون به صرفه یا جذاب بودن یک برنامه کاربردی برای مصرف‌کنندگان، ایده‌هایی وجود دارد

معیارهای ارزیابی عملکرد

- از کدام معیار استفاده کنیم؟
- معیارهای ارزیابی مختلفی می‌توانند برای اندازه‌گیری اثربخشی یک برنامه استفاده شوند
- این معیارها معمولاً با تابع ضرر مورد استفاده برای آموزش مدل متفاوت هستند
- گاهی اوقات مرتکب شدن یک نوع خطا از نوع دیگر پرهزینه‌تر است
 - سیستم تشخیص ایمیل هرزنامه
 - دسته‌بندی نادرست یک پیام قانونی به عنوان هرزنامه می‌تواند هزینه بسیار بیشتری داشته باشد



معیارهای ارزیابی عملکرد



- آموزش یک دسته‌بند دوسطحی برای تشخیص برخی رویدادهای نادر

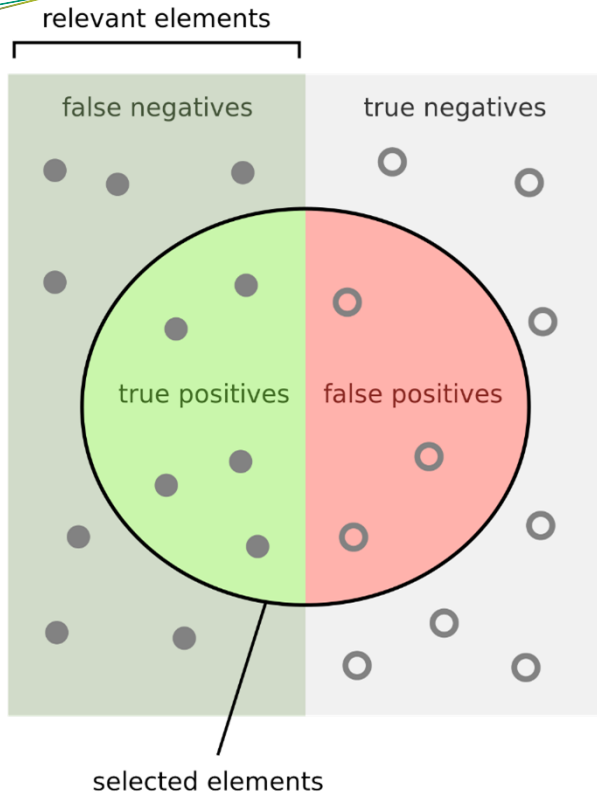
- آزمایش پزشکی برای یک بیماری نادر

- ۰.۱٪ احتمال ابتلا به بیماری

- دقت ۹۹.۹٪ با کدنویسی ساده!

- واضح است که دقت (Accuracy) معیار مناسبی برای توصیف عملکرد چنین مسائلی نیست

Recall و Precision



• Precision:

- درصد نمونه‌هایی که توسط مدل به عنوان کلاس مثبت تشخیص داده شده‌اند و درست بوده‌اند

$$Precision = \frac{TP}{TP + FP}$$

• Recall:

- درصد نمونه‌هایی که مثبت بوده‌اند و به درستی توسط مدل تشخیص داده شده‌اند

$$Recall = \frac{TP}{TP + FN}$$

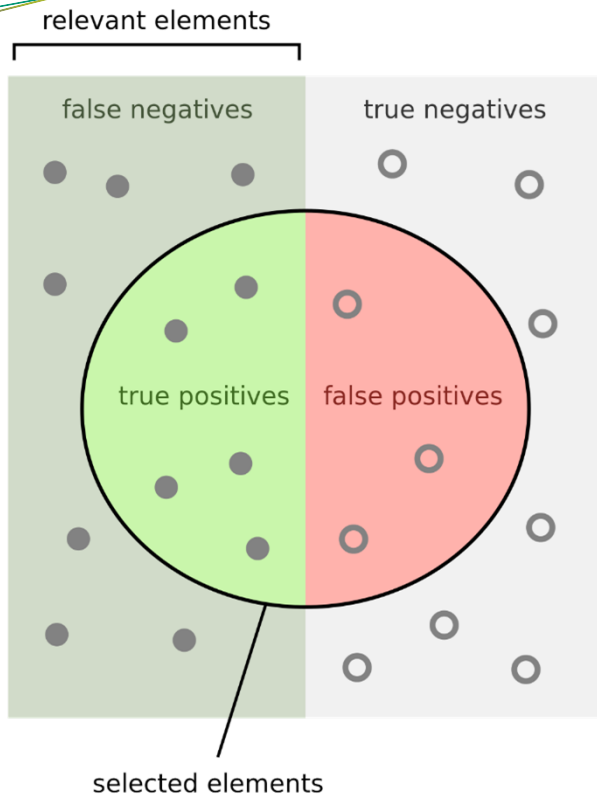
How many selected items are relevant?

$$Precision = \frac{\text{Green}}{\text{Green} + \text{Red}}$$

How many relevant items are selected?

$$Recall = \frac{\text{Green}}{\text{Green} + \text{Green}}$$

Recall و Precision



- یک مدل که می‌گوید هیچ کس این بیماری را ندارد
- Precision: کامل (۱۰۰٪)

- Recall: صفر

- یک مدل که می‌گوید تمام افراد این بیماری را دارند
- Recall: کامل (۱۰۰٪)

- Precision: برابر با درصد افراد مبتلا به این بیماری (مثلاً ۰.۰۰۱)

How many selected items are relevant?

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall و Precision

- خلاصه‌سازی PR با یک عدد

$$F_1 = \frac{2PR}{P + R}$$

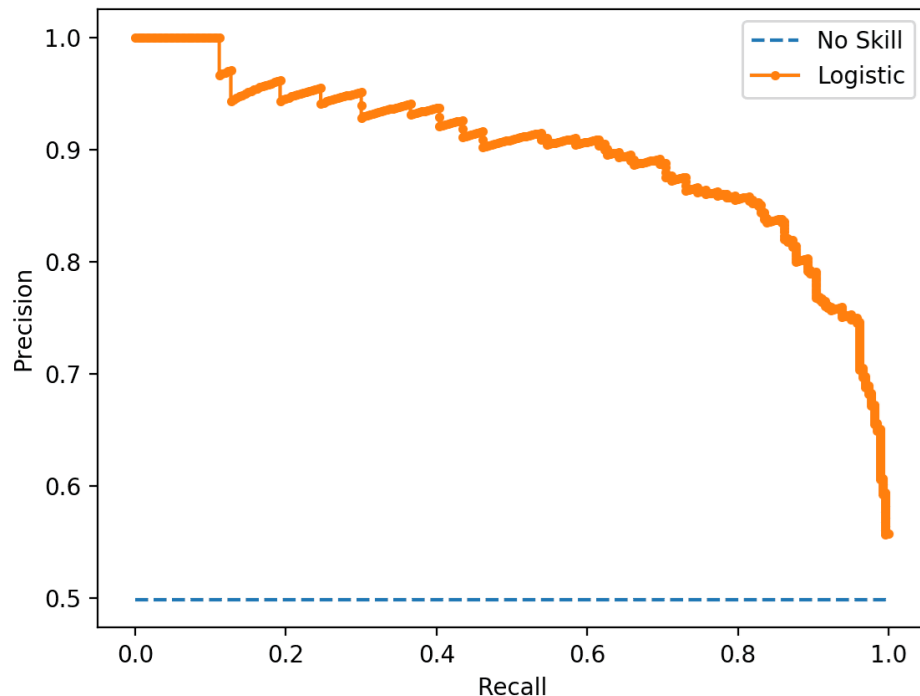
F-score -

- خلاصه‌سازی PR با گراف

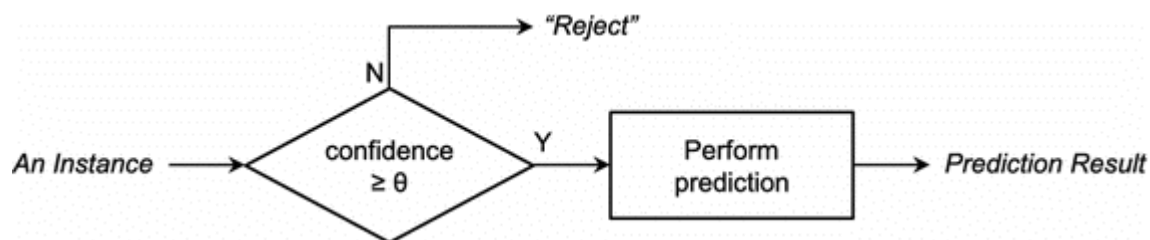
- Precision در محور y و Recall در محور x

- به طور معمول دسته‌بند برای هر کلاس یک امتیاز را محاسبه می‌کند و انتظار می‌رود کلاس درست دارای امتیاز بالاتری باشد

- سطح زیر منحنی (AUC) یکی از معیارهای بسیار پرکاربرد برای خلاصه‌سازی این منحنی است



معیارهای ارزیابی عملکرد



- امتناع از تصمیم‌گیری

- نیاز است تخمین بزنیم که چقدر در مورد تصمیم گرفته شده مطمئن هستیم
- به خصوص اگر یک تصمیم اشتباه می‌تواند ضرر زیادی داشته باشد و اگر یک اپراتور انسانی بتواند گاهی مواقع مسئولیت را بر عهده بگیرد

- کاهش میزان کاری که انسان باید انجام دهد

- پوشش (Coverage)

- محدوده پاسخ سیستم (برای چند درصد تصمیم نمی‌گیرد)

- در پروژه Street View هدف‌گذاری زیر انجام شده است

- دقت ۹۸٪ با پوشش ۹۵٪

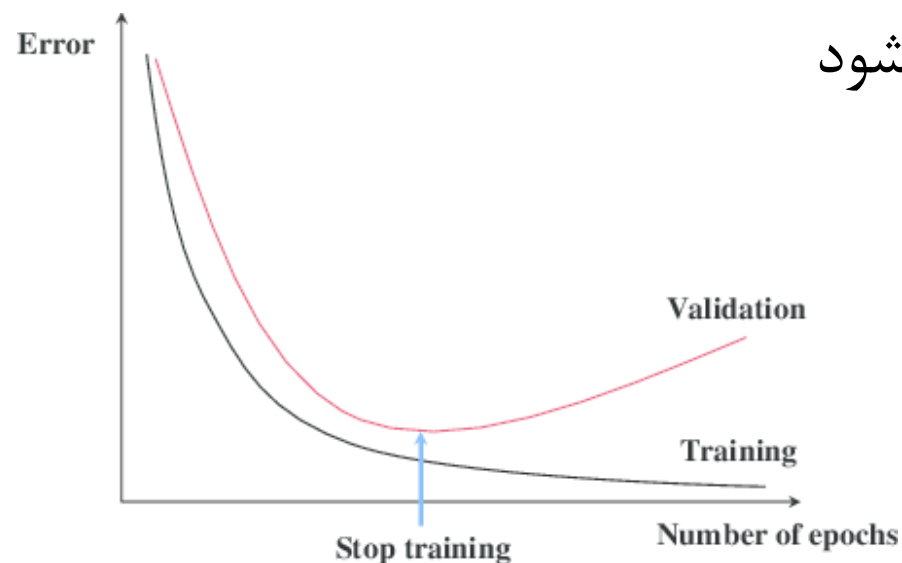


مدل‌های پایه پیش فرض

- یک سیستم end-to-end منطقی پیاده‌سازی کنید
 - یادگیری عمیق به عنوان یک رویکرد end-to-end با کدهای بسیار زیادی که در دسترس است، اجازه می‌دهد تا به راحتی یک الگوریتم پایه را پیاده‌سازی کنیم
- توصیه‌هایی برای انتخاب الگوریتم پایه
 - بسته به پیچیدگی مسئله
 - مدل یادگیری با عمق زیاد یا کم
 - بسته به ساختار داده‌ها
 - استفاده از لایه‌های کاملاً متصل و/یا لایه‌های کانولوشنی و/یا لایه‌های بازگشتی
- بهینه‌ساز Adam با مقادیر $\text{beta1} = 0.9$ و $\text{beta2} = 0.999$ و $\text{learning_rate} = 1\text{e-}3$ or $5\text{e-}4$ یک نقطه شروع خوب است

مدل‌های پایه پیش فرض

- اگر مجموعه آموزشی محدود است
 - می‌توان از همان ابتدا از روش‌های منظم‌سازی استفاده کرد
 - Dropout، داده‌افزایی، منظم‌سازی وزن‌ها و ...
 - نرمال‌سازی می‌تواند منجر به بهبود تعمیم‌دهی شود
- توقف زودهنگام (Early Stopping) تقریباً به طور کلی باید استفاده شود
- اگر مسئله مورد نظر شبیه به مسئله شناخته شده دیگری است
 - مدل و الگوریتمی که در حال حاضر بهترین عملکرد را دارد کپی کنید
 - از وزن‌های پیش‌آموخته به درستی استفاده کنید

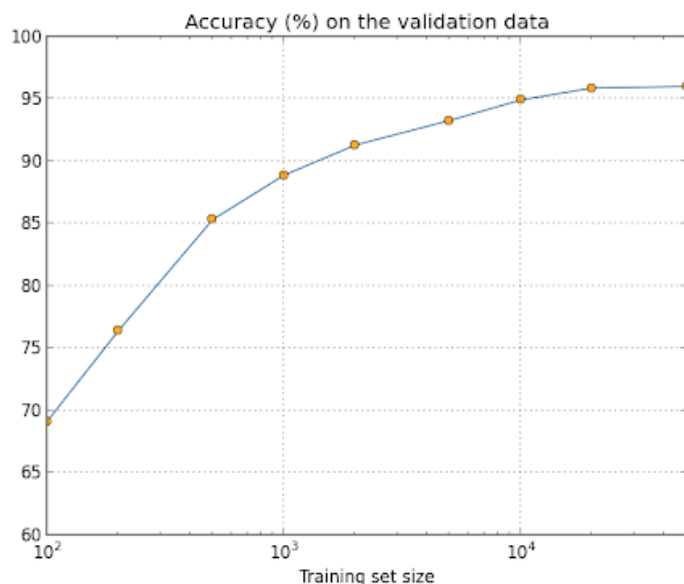


جمع‌آوری داده‌های بیشتر

- چگونه می‌توان تصمیم گرفت که داده‌های بیشتری را جمع‌آوری کرد؟
 - آیا عملکرد روی مجموعه آموزشی قابل قبول است یا خیر؟
- اگر عملکرد در مجموعه آموزشی ضعیف باشد
 - دلیلی برای جمع‌آوری داده‌های بیشتر وجود ندارد
 - ظرفیت مدل را افزایش دهید
 - الگوریتم یادگیری را بهبود دهید
- اگر مدل‌های قوی به خوبی کار نمی‌کنند
 - ممکن است مشکل از کیفیت داده‌های آموزشی باشد
 - سعی کنید داده‌های تمیزتر جمع‌آوری کنید یا مجموعه کامل‌تری از ویژگی‌ها را جمع‌آوری کنید

جمع‌آوری داده‌های بیشتر

- اگر عملکرد در مجموعه آزمون قابل قبول است
 - نیاز به کار دیگری نیست!
- اگر عملکرد در مجموعه آزمون بسیار بدتر از مجموعه آموزشی
 - جمع‌آوری داده‌های بیشتر یکی از موثرترین راه‌حل‌ها است
- ملاحظات کلیدی
 - هزینه و امکان‌سنجی جمع‌آوری داده‌های بیشتر
 - هزینه و امکان‌سنجی کاهش خطای آزمون با روش‌های دیگر
 - تخمین مقدار داده‌ای که انتظار می‌رود ضروری باشد



انتخاب ابرپارامترها

- در طراحی و آموزش هر شبکه ابرپارامترهای بسیار زیادی از جمله نرخ آموزش، تعداد لایه‌ها، تعداد واحدها، توابع فعال‌سازی، تابع ضرر، ابعاد فیلترها و ... وجود دارند

- تاثیر بر زمان و حافظه اجرای الگوریتم

- تاثیر بر کیفیت مدل آموزش‌دیده

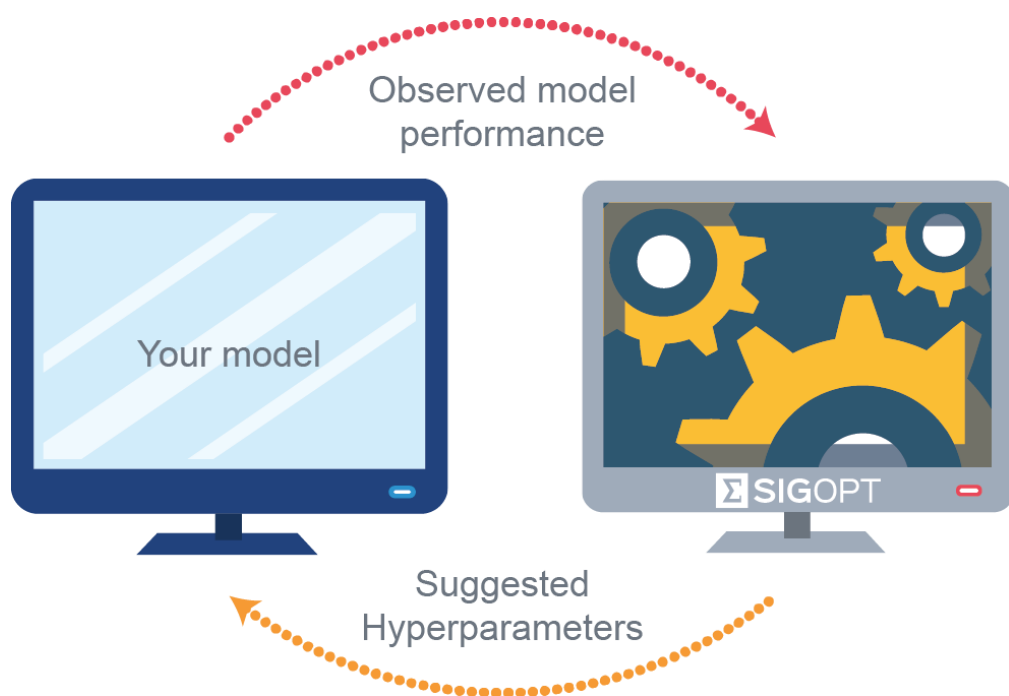
- دو رویکرد اصلی برای انتخاب ابرپارامترها وجود دارد:

- انتخاب دستی

- نیاز به درک ابرپارامترها دارد

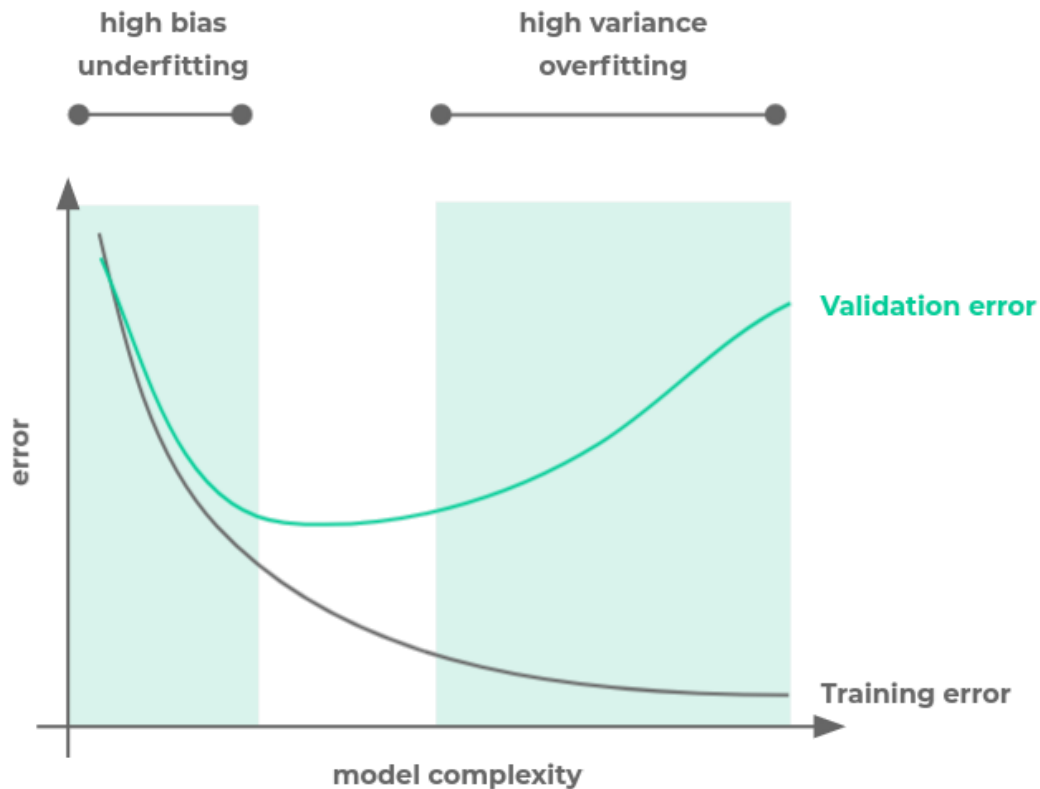
- انتخاب خودکار

- اغلب از نظر محاسباتی بسیار پرهزینه هستند



تنظیم دستی ابرپارامترها

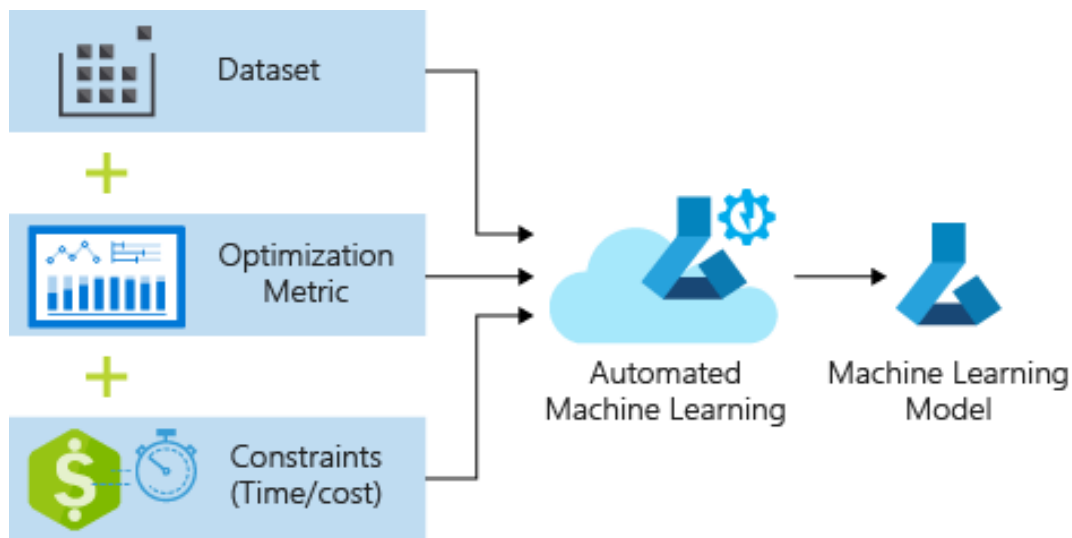
- نیاز است تا رابطه میان ابرپارامترها با خطای آموزش، خطای تعمیم‌دهی، و منابع محاسباتی (زمان و حافظه) درک شده باشد



ابریارمتر	ظرفیت مدل افزایش می‌یابد اگر ...
تعداد لایه‌های میانی	
ضریب کاهش وزن	
ابعاد کرنل کانولوشنی	
نرخ Dropout	
نرخ آموزش	

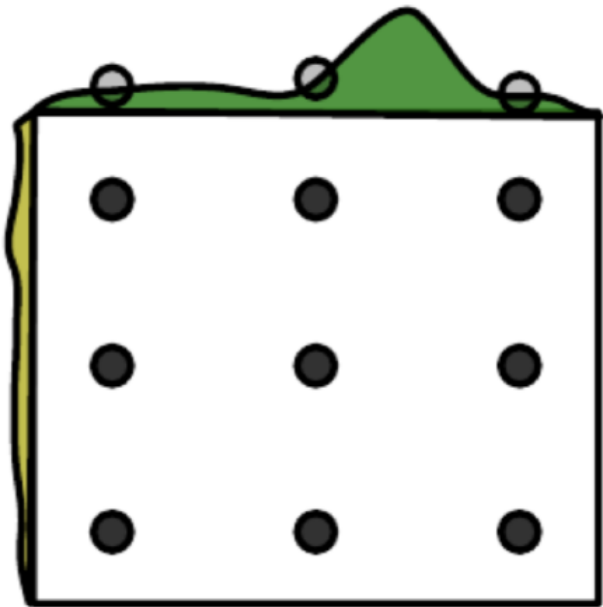
تنظیم خودکار ابرپارامترها

- تنظیم دستی ابرپارامترها می‌تواند بسیار خوب کار کند اگر کاربر دارای تجربه کافی باشد و نقطه شروع خوبی داشته باشد
- الگوریتم یادگیری ماشین ایده‌آل فقط یک مجموعه داده و اهداف را می‌گیرد و تابع تبدیل را بدون نیاز به تنظیم دستی ابرپارامترها محاسبه می‌کند



جستجوی شبکه‌ای

- اگر تعداد ابرپارامترها سه یا کمتر باشد
- برای هر ابرپارامتر، مجموعه محدودی از مقادیر برای جستجو انتخاب می‌شود
 - مانند $lr \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$
- هزینه محاسباتی بالا
 - $O(n^m)$
- معمولاً زمانی بهترین عملکرد را دارد که در چندین مرحله انجام شود
 - در هر مرحله محدوده جستجو کوچک‌تر و دقیق‌تر می‌شود



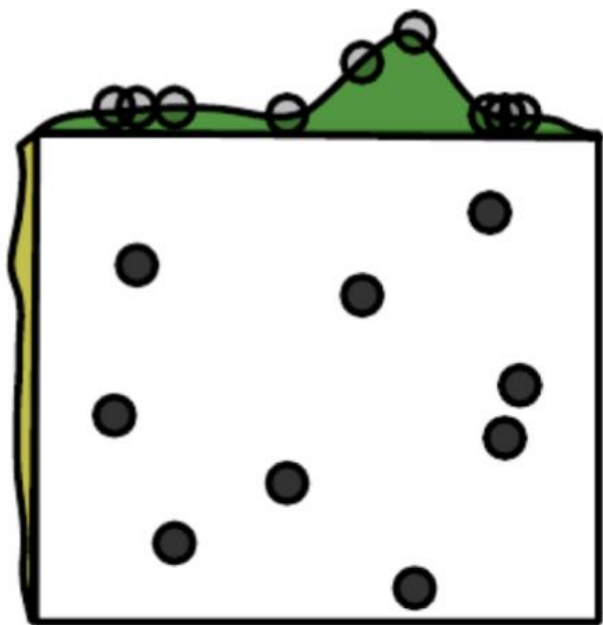
جستجوی تصادفی

- پیاده‌سازی ساده، و همگرایی سریع‌تر
- برای هر ابرپارامتر یک توزیع تصادفی تعریف می‌شود

$$\log(lr) \sim u(-1, -5) -$$

$$lr = 10^{\log(lr)} -$$

- مجموعه بزرگتری از مقادیر جستجو می‌شود
- اگر برخی ابرپارامترها اثر زیادی نداشته باشند، بسیار کارآمدتر است



بهینه‌سازی ابرپارامتر مبتنی بر مدل

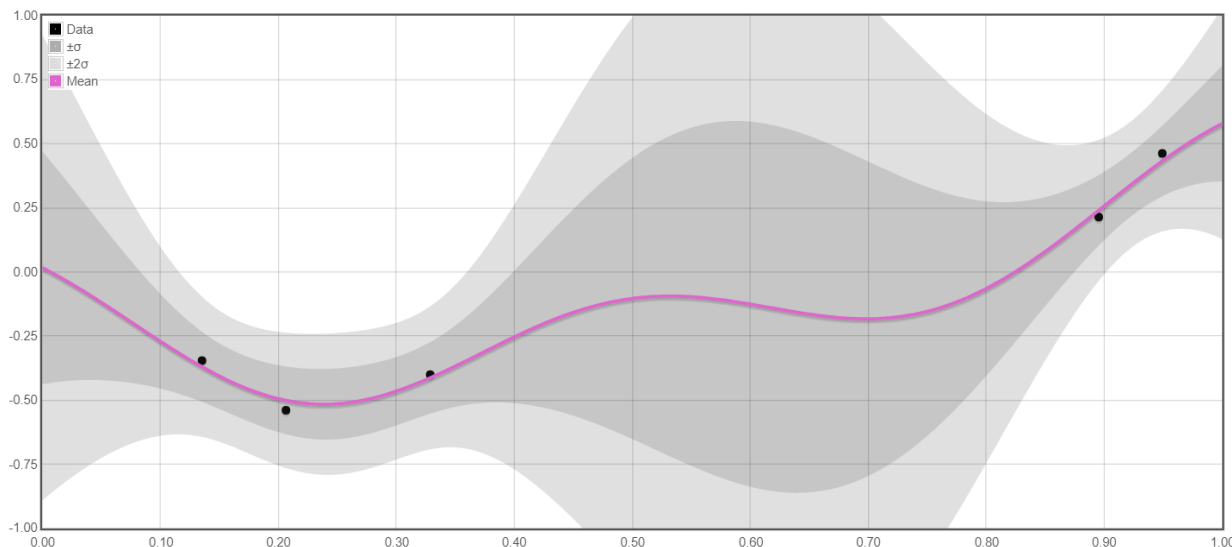
- جستجوی ابرپارامترهای مناسب را می‌توان به عنوان یک مسئله بهینه‌سازی در نظر گرفت
 - متغیرهای بهینه‌سازی، ابرپارامترها هستند
 - تابعی که باید بهینه شود خطای مجموعه اعتبارسنجی است
 - متأسفانه، در اکثر موارد گرادیان در دسترس نیست

$$A^* = \arg \min_{A \in \mathcal{A}} \mathcal{L}_{val}(A(w^*), D_{val})$$

$$s. t. \quad w^* = \arg \min_w \mathcal{L}_{train}(A(w), D_{train})$$

بهینه‌سازی ابرپارامتر مبتنی بر مدل

- می‌توان بر اساس آزمایش‌های انجام شده، مدلی برای خطای مجموعه اعتبارسنجی ایجاد کرد
 - با انجام بهینه‌سازی در این مدل، ابرپارامترهای جدیدی پیشنهاد خواهند شد
- می‌توان از یک مدل رگرسیون Bayesian برای تخمین مقدار مورد انتظار و همچنین عدم قطعیت آن به ازای هر ابرپارامتر استفاده کرد
- برای این بهینه‌سازی باید تعادلی ایجاد شود میان:
 - اکتشاف (Exploration)
 - بهره‌برداری (Exploitation)



<http://chifeng.scripts.mit.edu/stuff/gp-demo/>



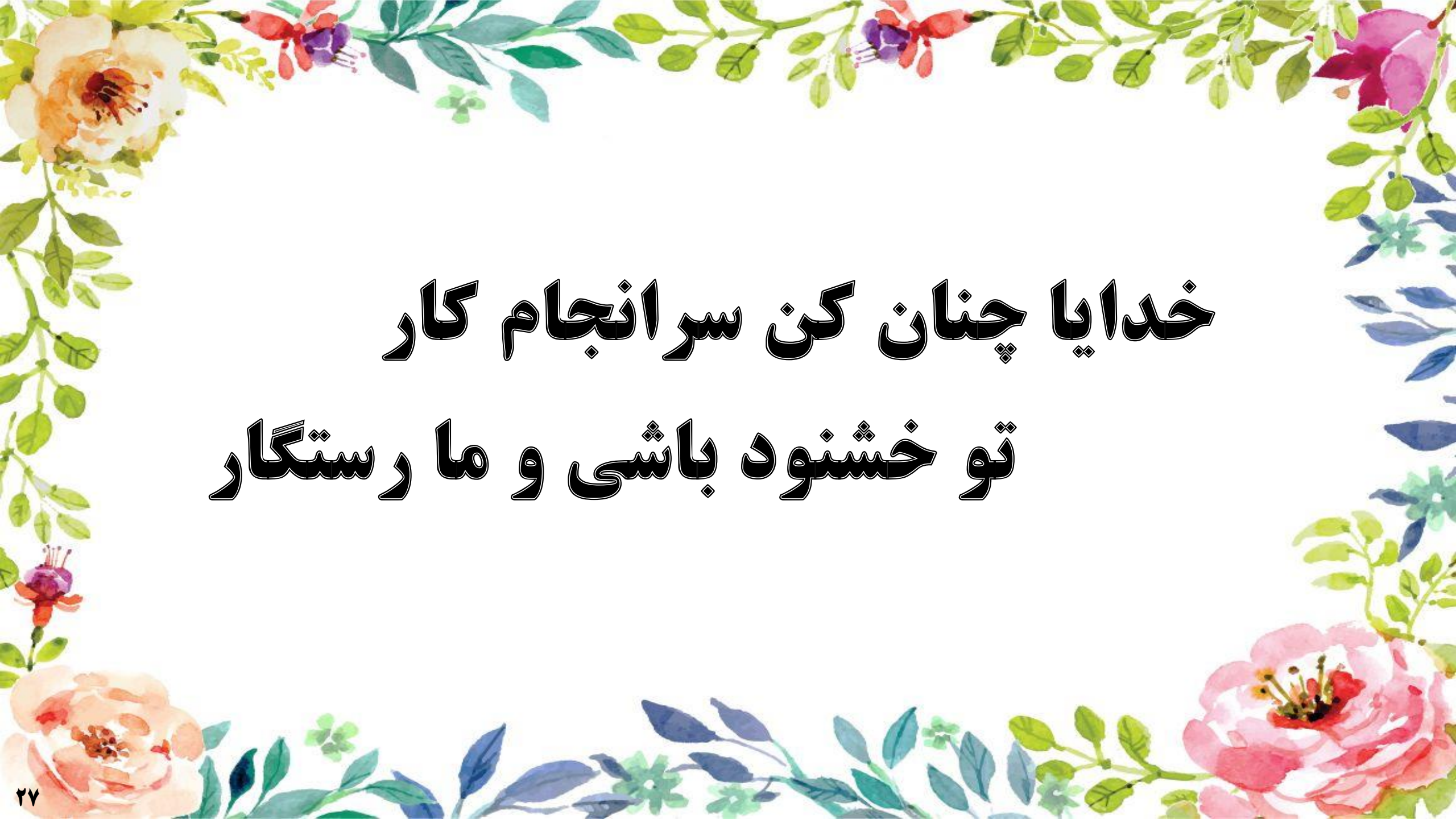
AutoKeras: An AutoML system based on Keras. It is developed by **DATA Lab** at Texas A&M University. The goal of AutoKeras is to make machine learning accessible to everyone.

[AutoKeras](#) •

[KerasTuner](#) •

KerasTuner

KerasTuner is an easy-to-use, scalable hyperparameter optimization framework that solves the pain points of hyperparameter search. Easily configure your search space with a define-by-run syntax, then leverage one of the available search algorithms to find the best hyperparameter values for your models. KerasTuner comes with Bayesian Optimization, Hyperband, and Random Search algorithms built-in, and is also designed to be easy for researchers to extend in order to experiment with new search algorithms.



خدایا چنان کن سرانجام کار
تو خشنود باشی و ما رستگار