

NAS (Neural Architecture Search):

جستجوی ساختار شبکه‌های عصبی (Neural Architecture Search یا NAS) یک روش برای خودکارسازی فرآیند طراحی شبکه‌های عصبی است. در این روش، یک الگوریتم به صورت خودکار ساختار شبکه‌های عصبی را تولید می‌کند، سپس شبکه‌های تولید شده برای مسئله داده شده آموزش داده می‌شوند و بهترین ساختار به عنوان نتیجه انتخاب می‌شود. یکی از روش‌های NAS، الگوریتم تکاملی است که در آن جمعیتی از شبکه‌های عصبی با ساختارهای مختلف تولید می‌شود و سپس با استفاده از عملگرهای تکاملی مانند تلاش برای بقا، تولید نسل جدیدی از شبکه‌ها انجام می‌شود. این فرآیند تکرار می‌شود تا به یک ساختار بهینه برای مسئله داده شده برسیم. روش‌های دیگری نیز برای جستجوی ساختار شبکه‌های عصبی وجود دارد، به عنوان مثال الگوریتم‌های مبتنی بر گراف، بهینه‌سازی گرادیان و روش‌های تولید تصادفی شبکه‌های عصبی. یکی از مزیت‌های استفاده از NAS، کاهش نیاز به دانش دقیق در زمینه طراحی شبکه‌های عصبی است. با استفاده از NAS، می‌توان به صورت خودکار و بدون نیاز به دانش خاص، ساختار بهینه برای شبکه‌های عصبی را پیدا کرد. الگوریتم‌هایی مانند Reinforcement learning و روش‌های مبتنی بر گرادیان زمان بسیر زیادی را صرف می‌کنند. یک ایده برای کاهش زمان بر بودن استفاده از Network Morphism هست. Network Morphism یک تکنیک در حوزه یادگیری عمیق است که برای تغییر و اصلاح معماری شبکه‌های عصبی به صورت خودکار به کار می‌رود. با استفاده از این تکنیک، می‌توانیم شبکه‌های عصبی را به صورت خودکار با استفاده از عملیاتی مانند افزودن لایه، حذف لایه و یا تغییر اتصالات، تغییر دهیم. یک مثال از استفاده از تکنیک Network Morphism، استفاده از شبکه‌های عصبی متفاوت برای تشخیص تصاویر است. در این مثال، ما می‌توانیم با استفاده از تکنیک Network Morphism، شبکه‌های عصبی مختلفی با ساختار متفاوت ولی با عملکرد مشابه به دست بیاوریم. به این ترتیب، می‌توانیم معماری شبکه‌های عصبی را به صورت خودکار تغییر داده و به جای آزمایش شبکه‌های عصبی مختلف یکی یکی، از تکنیک Network Morphism برای بهبود عملکرد شبکه‌های عصبی استفاده کنیم.

:Auto keras

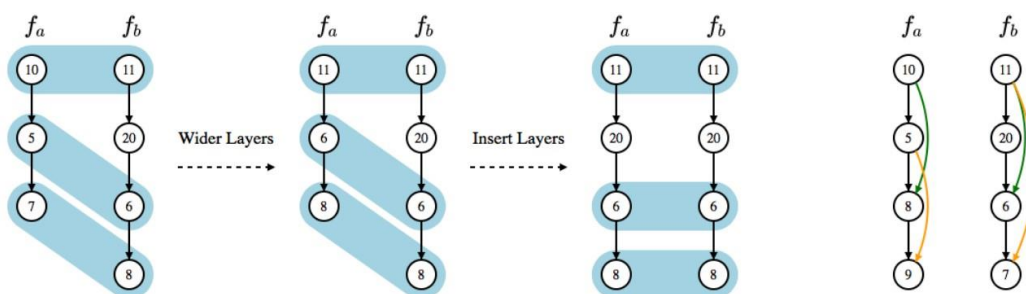
AutoKeras یک کتابخانه یادگیری عمیق (Deep Learning Library) برای Python است که توسط شرکت Data Analytics at Texas A&M University توسعه داده شده است. AutoKeras به شما این امکان را می‌دهد که با استفاده از یادگیری خودکار، مدل‌های شبکه عصبی پیچشی (Convolutional Neural Networks)، شبکه عصبی بازگشتی (Recurrent Neural Networks) و شبکه عصبی چند لایه (Multi-Layer Perceptron) را برای مسائل دسته‌بندی، پیش‌بینی و رگرسیون، بدون نیاز به دانش تخصصی در زمینه یادگیری عمیق، آموزش دهید.

AutoKeras از الگوریتم‌های بهینه‌سازی جستجوی فضای جستجوی خودکار (Automated Neural Architecture Search) برای پیدا کردن بهترین مدل‌های شبکه عصبی در فضای جستجو استفاده می‌کند. با استفاده از AutoKeras، شما نیازی به انجام آزمایش‌های آزمایشی مختلف برای به دست آوردن بهترین مدل‌های شبکه عصبی ندارید. به جای آن، AutoKeras به‌طور خودکار مدل‌های مختلف را آزمایش می‌کند و بهترین مدل را برای شما پیدا می‌کند

AutoKeras از پیاده‌سازی TensorFlow و Keras برای پیاده‌سازی شبکه‌های عصبی استفاده می‌کند. AutoKeras همچنین از پیاده‌سازی یک الگوریتم خاص برای تنظیم پارامترها (Hyperparameter Tuning) با نام Bayesian Optimization برای بهبود عملکرد مدل استفاده می‌کند.

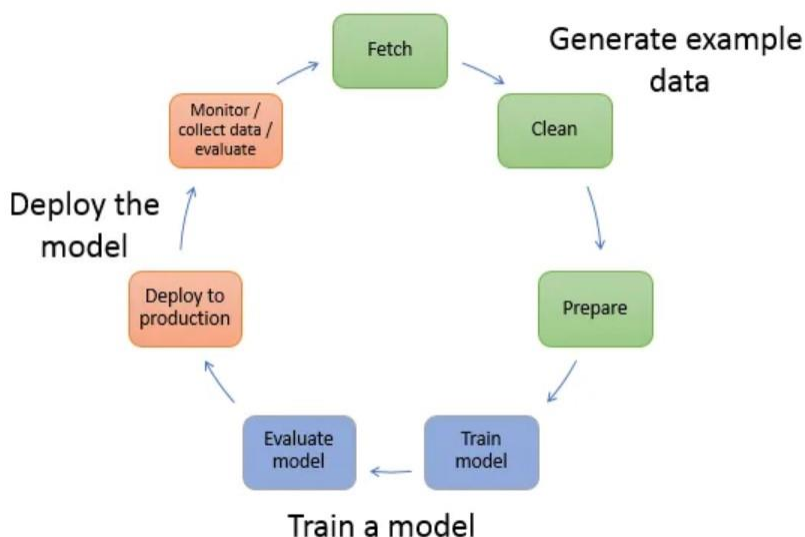
نحوه پیاده سازی Auto keras:

Auto keras با استفاده از Bayesian optimization و هم چنین شبکه های مورفیسیم به جستجو در فضای شبکه های عصبی میپردازد. در ابتدا نیاز هست که کمی درباره Bayesian optimization صحبت بکنیم و چگونه در پیاده سازی Auto keras استفاده میشه یکی از چالش های ما این هست که نمی تونیم شبکه عصبی را به فضای اقلیدسی مدل بکنیم به همین علت می توانیم ان را به یک گراف مدل بکنیم می توانیم یک گراف تعریف بکنیم که هر گره در گراف معادل با یک معماری شبکه عصبی است و هر یال معادل با یک عمل مورفیسیم است که می تواند شامل اضافه کردن یک لایه، حذف یک لایه، تغییر تعداد فیلترها و یا تغییر توابع فعال سازی باشد. به طور کلی، در هر مرحله از بهینه سازی بیزی، با استفاده از تابع تکیه و تخمینات Gaussian Process، یک گره جدید در گراف پیشنهاد می شود. سپس، ساختار شبکه عصبی معادل با این گره تولید می شود و عملکرد آن با استفاده از تابع هزینه محاسبه می شود. سپس، این اطلاعات به الگوریتم بهینه سازی بیزی بازگردانده می شود تا در مرحله بعد، با توجه به این اطلاعات، گره جدیدی در گراف پیشنهاد شود. به طور خلاصه، هدف بهینه سازی بیزی در شبکه های عصبی، پیدا کردن ساختار شبکه عصبی بهینه است که بیشترین بهبود در عملکرد شبکه را دارد، با استفاده از تابع تکیه و روش های Gaussian Process. هم چنین یک مفهوم هم به نام Distance Neural Network Kernel که به صورت زیر تعریف میشود:



ویژگی Auto model

در ابتدا کمی درباره مدل های یادگیری عمیق صحبت میکنم که چه مرحله ای را طی میکنند در ابتدا داده ها را fetch می کنیم و بعد clean می کنیم و آماده یادگیری می کنیم و در انتها ان ها را ارزیابی می کنیم :



Auto keras خیلی از لحاظ فرمت ورودی هم انعطاف پذیر است می تواند تصاویر یک کانال یا سه کانال دریافت بکنه. در واقع در Auto keras میتوانیم هر عکسی با هر تعداد کانال بدیم. و هنگامی که از Auto keras استفاده میکنیم خیلی از پیش پردازش های که باید روی دیتاست انجام بدیم به صورت خودکار انجام شده.

Auto keras از الگوریتم های بهینه سازی نظیر Adam و SGD برای بهینه سازی شبکه های پیدا شده استفاده می کند.

AutoKeras قادر است به صورت خودکار پارامترهای مختلف شبکه های عصبی را جستجو کند تا بهترین مدل را پیدا کند. در جستجوی پارامتر، AutoKeras می تواند مجموعه ای از پارامترهای مختلف را سرچ کند که اغلب شامل موارد زیر است:

۱. تعداد لایه‌های مخفی شبکه، تعداد نورون‌ها در هر لایه و نوع هر لایه، مانند لایه‌های کاملاً متصل، لایه‌های پیچشی و لایه‌های بازگشتی.

۲. نرخ یادگیری (Learning Rate) که میزان تأثیر تغییرات در وزن‌های شبکه را تنظیم می‌کند.

۳. تعداد دوره‌های آموزش (Epochs) که تعداد بارهایی است که شبکه روی داده‌های آموزش آموزش داده می‌شود.

۴. اندازه دسته‌ها (Batch Size) که تعداد نمونه‌هایی است که در هر دوره آموزش به شبکه داده می‌شود.

۵. تعداد ورودی‌های شبکه، انواع ورودی‌ها و هر پارامتر دیگری که برای شبکه مشخص شده است.

AutoKeras با استفاده از یک الگوریتم جستجوی تصادفی مبتنی بر گوگل کراوفورد (Google's Vizier algorithm) بهترین مقادیر برای این پارامترها را پیدا می‌کند. همچنین با استفاده از روش‌های هوشمند ترکیبی، AutoKeras می‌تواند برای هر مسئله خاص، بهترین معماری شبکه را پیشنهاد دهد.

AutoModel یک کلاس در کتابخانه AutoKeras است که به شما این امکان را می‌دهد که به‌طور دلخواه یک مدل شبکه عصبی را ایجاد کنید و آن را با استفاده از الگوریتم جستجوی خودکار AutoKeras آموزش دهید. با استفاده از کلاس AutoModel، شما می‌توانید به‌طور دقیق‌تر فضای جستجوی خود را تنظیم کنید و مدل‌هایی با یادگیری عمیق و دقت بالا ایجاد کنید.

برای استفاده از کلاس AutoModel، ابتدا باید یک شی از کلاس ImageInput ایجاد کنید تا تصاویر ورودی را دریافت کند. سپس شما می‌توانید از بلوک‌های مختلفی مانند ImageBlock، TextBlock و StructuredDataBlock استفاده کنید تا مدل شبکه عصبی خود را ایجاد کنید. هر بلوک شامل یک معماری شبکه عصبی مشخص است که برای جستجو در فضای جستجو استفاده می‌شود.

با استفاده از کلاس AutoModel، شما می‌توانید به‌صورت دقیق‌تر و جزئی‌تر فضای جستجوی خود را تنظیم کنید. برای مثال، شما می‌توانید پارامترهای مختلفی مانند تعداد لایه‌ها، تعداد نورون‌ها و تعداد پارامترها را به‌طور دقیق مشخص کنید و به‌این ترتیب، مدل شبکه عصبی خود را بهترین شکل ممکن طراحی کنید.

من Auto keras را روی دیتاست cifar10 نیز امتحان کردم.

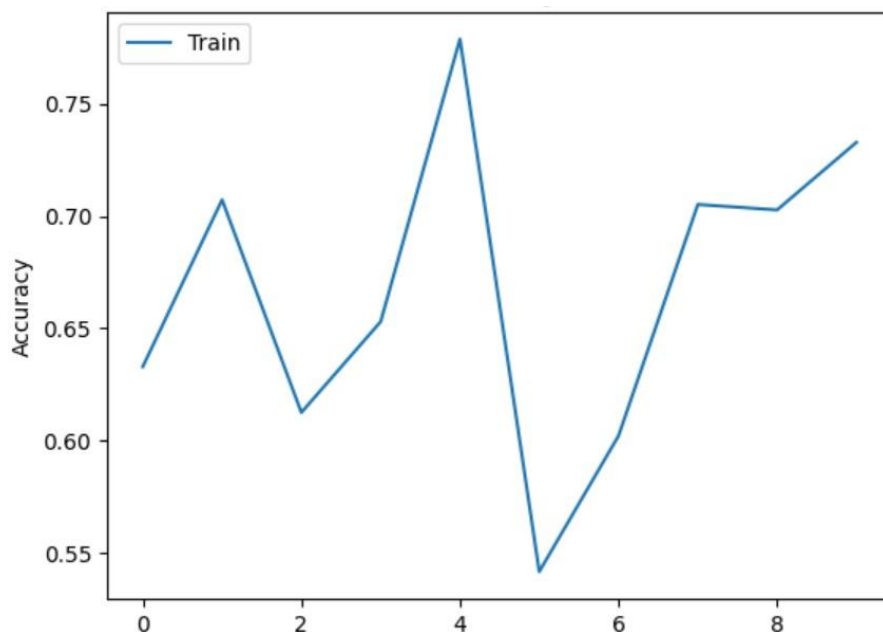
در قسمت اول کد، یک مدل `AutoKeras ImageClassifier` تعریف می‌کنم که شامل یک شبکه عصبی پیچشی برای دسته‌بندی تصویر است. سپس با استفاده از تابع `accuracy_score` از کتابخانه `scikit-learn`، دقت مدل بر روی داده‌های تست محاسبه می‌شود.

در قسمت بعدی کد، یک شبکه عصبی پیچشی از نوع `Inception` برای دسته‌بندی تصویر با استفاده از `Keras` تعریف شده است. این شبکه شامل چهار بخش مختلف است که هر یک با استفاده از لایه‌های مختلفی از جمله `Conv2D` و `MaxPooling2D` تعریف شده‌اند.

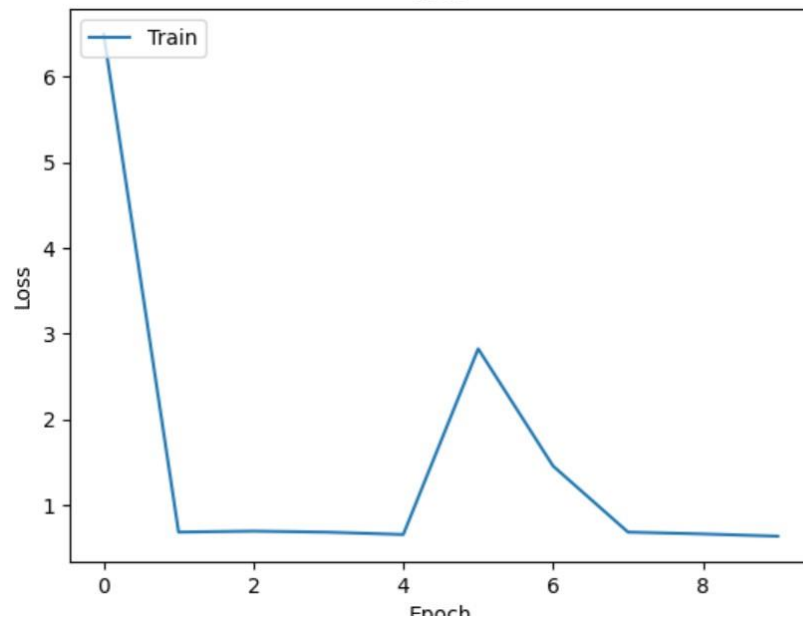
در قسمت بعدی کد، یک شبکه عصبی پیچشی `Inception` از نوع `AutoKeras` برای دسته‌بندی تصویر با استفاده از داده‌های `Cifar-10` تعریف شده است. در این شبکه، با استفاده از تابع `ImageInput` ورودی تصویر تعریف شده است. سپس با استفاده از توابع `Normalization` و `ImageAugmentation`، تصاویر ورودی به شبکه پیش‌پردازش می‌شوند. سپس با استفاده از تابع `ResNetBlock` و لایه `ClassificationHead`، شبکه عصبی تعریف می‌شود. در نهایت، با استفاده از تابع `fit`، مدل آموزش داده می‌شود.

کد در ریپو گیت هاب گذاشته شده است.

بر روی دیتاست انجیر نیز `train` کردم و نتایج به صورت زیر شد:



و برای `loss` نیز نتایج به صورت زیر شد:



ولی اگر فقط فضای سرچ رو محدود به بلاک های resnet بکنیم هم زمانی که صرف train میشود خیلی کمتر است و هم نتایج loss, accuracy خیلی بهتر شد:

