

بایاس

بایاس یک پارامتر اضافی در شبکه عصبی است که به هر نورون اضافه می‌شود. نقش بایاس شبیه به ثابت در یک معادله خطی است. کارکرد اصلی بایاس به شرح زیر است:

- افزایش انعطاف‌پذیری: بایاس به مدل امکان می‌دهد تا تابع تصمیم‌گیری خود را به گونه‌ای تنظیم کند که بهتر بتواند با داده‌های ورودی مختلف کنار بیاید. به عبارت دیگر، بایاس به مدل کمک می‌کند تا خروجی‌های دقیق‌تری را حتی زمانی که همه ورودی‌ها صفر هستند، ارائه دهد.
- کنترل نقطه عملکرد: بایاس به تغییر مکان نقطه عملکرد تابع فعال‌ساز کمک می‌کند. این امر به نورون اجازه می‌دهد تا فعال یا غیرفعال شود بر اساس نیاز بهینه‌سازی مدل.

تابع فعال‌ساز

تابع فعال‌ساز نقش مهمی در شبکه‌های عصبی دارد و به مدل اجازه می‌دهد تا پیچیدگی‌های بیشتری را در داده‌ها یاد بگیرد.

- معرفی غیرخطی بودن: توابع فعال‌ساز غیرخطی مانند ReLU، سیگموئید، و تانژانت هیپربولیک به شبکه اجازه می‌دهند تا روابط غیرخطی بین ورودی‌ها و خروجی‌ها را یاد بگیرد. این قابلیت برای مدل‌سازی اکثر مسائل واقعی که معمولاً غیرخطی هستند ضروری است.
- کمک به همگرایی: برخی توابع فعال‌ساز، مانند ReLU، در تسریع فرایند آموزش شبکه‌های عصبی عمیق تأثیر مثبت دارند به دلیل اینکه به مدیریت مشکل ناپدید شدن یا انفجار گرادیان‌ها کمک می‌کنند.
- اجازه به مدل برای تعمیم دادن: استفاده از توابع فعال‌ساز مناسب می‌تواند به مدل کمک کند تا بهتر بر روی داده‌های دیده نشده تعمیم یابد، که این یکی از اهداف اصلی در یادگیری ماشین است.

در مجموع، بایاس و تابع فعال‌ساز هر دو به شبکه‌های عصبی کمک می‌کنند تا انعطاف‌پذیری و قابلیت تعمیم دهی بیشتری داشته باشند، و بدون آن‌ها، شبکه‌های عصبی نمی‌توانند فراتر از توابع ساده خطی عمل کنند و نمی‌توانند مسائل پیچیده‌تر را حل کنند.

2

قاعده یادگیری

- **پرسپترون:** پرسپترون از قاعده یادگیری پرسپترون استفاده می‌کند که وزن‌ها را تنها زمانی به‌روز می‌کند که یک خطا رخ دهد (یعنی وقتی که طبقه‌بندی اشتباه است). این به‌روزرسانی وزن‌ها به صورت گسسته و بر اساس خطاهای طبقه‌بندی صورت می‌گیرد.
- **آدالاین:** آدالاین از قاعده به‌روزرسانی وزن‌ها استفاده می‌کند که خطای محاسبه شده بر اساس تفاوت بین خروجی مورد انتظار و خروجی محاسبه شده (قبل از اعمال تابع فعال‌سازی) است. این رویکرد به این معنی است که آدالاین می‌تواند در یادگیری خطی موثرتر باشد و از قاعده کاهش مجذور خطا (LMS) برای بهینه‌سازی استفاده می‌کند.

همگرایی

- **پرسپترون:** قاعده یادگیری پرسپترون تضمین می‌کند که اگر داده‌ها قابل جداسازی خطی باشند، الگوریتم پس از تعداد محدودی از تکرار به یک حل مناسب همگرا می‌شود.
 - **آدالاین:** آدالاین نیز تنها در شرایطی به یک حل مناسب همگرا می‌کند که داده‌ها قابل جداسازی خطی باشند، اما از آنجا که خطای مربعی را بهینه‌سازی می‌کند، این فرآیند ممکن است نرم‌تر و پایدارتر از پرسپترون باشد.
- این تفاوت‌ها باعث می‌شوند که آدالاین برای مسائلی که نیاز به تخمین دقیق‌تر و بهینه‌سازی پیوسته دارند، مناسب‌تر باشد، در حالی که پرسپترون برای مسائل ساده‌تر طبقه‌بندی که تنها نیاز به جداسازی داده‌ها دارند، کاربردی است.

الگوریتم بهینه‌سازی آدام Adam، (که مخفف عبارت Adaptive Moment Estimation است) یکی از روش‌های محبوب و پرکاربرد در زمینه یادگیری عمیق و بهینه‌سازی در هوش مصنوعی است. این الگوریتم برای اولین بار در سال 2015 معرفی شد و به سرعت به دلیل توانایی بالا در همگرایی سریع و کارایی بالا در مواجهه با مشکلات بهینه‌سازی بزرگ و پیچیده، محبوبیت یافت.

اصول کارکرد آدام

آدام یک روش بهینه‌سازی تکراری است که برای یافتن نقاط اکسترمم محلی در توابع هزینه بسیار غیرخطی و پیچیده‌ای که در یادگیری عمیق با آن‌ها مواجه هستیم، استفاده می‌شود. این الگوریتم با ترکیب مفاهیم موجود در الگوریتم‌های بهینه‌سازی RMSprop و AdaGrad، توانسته است عملکرد بهتری ارائه دهد.

مؤلفه‌های کلیدی آدام

1. **تخمین لحظه‌ای اول و دوم گرادیان:** آدام برای هر پارامتر دو مقدار تجمعی را محاسبه می‌کند. یکی برای میانگین متحرک نمایی (exponential moving average) گرادیان‌ها (موسوم به میانگین لحظه‌ای اول) و دیگری برای میانگین متحرک نمایی مربعات گرادیان‌ها (موسوم به میانگین لحظه‌ای دوم).
2. **تصحیح پیش‌فرض‌ها:** به دلیل این که میانگین‌های متحرک در ابتدای تکرارها ممکن است به سمت صفر متمایل شوند، آدام شامل یک مکانیزم تصحیح پیش‌فرض است که این میانگین‌ها را تعدیل می‌کند تا دقت بیشتری حاصل شود.
3. **به‌روزرسانی پارامترها:** در هر تکرار، پارامترها بر اساس میانگین‌های تصحیح شده لحظه‌ای به‌روزرسانی می‌شوند:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

Where:

- θ_t represents the current parameters,
- η is the learning rate,
- \hat{m}_t and \hat{v}_t are the bias-corrected first and second moment estimates respectively,
- ϵ is a small number to prevent division by zero.

مزایای آدام

- **همگرایی سریع:** آدام به سرعت به نتایج بهینه نزدیک می‌شود، که این برای مسائل پیچیده یادگیری عمیق بسیار مفید است.
- **کارآمد در مسائل بزرگ:** برای داده‌ها و مدل‌های بزرگ که نیازمند بهینه‌سازی هستند، آدام بسیار مؤثر عمل می‌کند.
- **خودتنظیم‌کننده:** تنظیم پارامترها در آدام کمتر نیاز به دخالت دستی دارد، زیرا این الگوریتم به طور خودکار تنظیماتی مانند نرخ یادگیری را انجام می‌دهد.