

پروژه چهارم – بهینه سازی

تا به حال همیشه از Gradient Descent برای به روزرسانی پارامترها و به حداقل رساندن هزینه استفاده کرده اید. در این پروژه، روش های بهینه سازی پیشرفته تری را می آموزید که می تواند سرعت یادگیری را افزایش دهد و حتی ممکن است شما را به یک مقدار نهایی بهتر برای هزینه برساند.

مرحله صفر: اضافه کردن کتابخانه های لازم

محتویات پوشه Requirements را در مسیر پروژه اضافه کنید.

مرحله اول: Gradient Descent

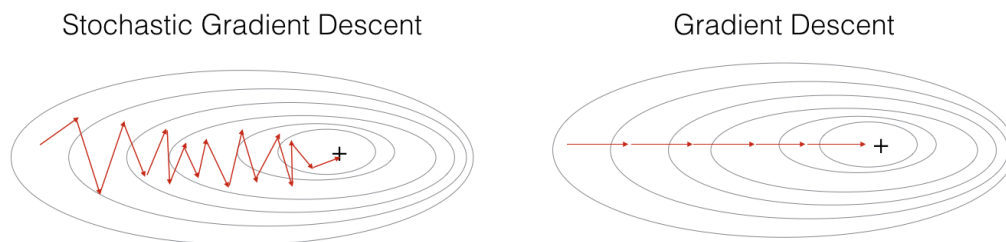
یک روش ساده بهینه سازی در یادگیری ماشین ، گرادیان نزولی (GD) است. هنگامی که تعداد نمونه ها در هر مرحله m باشد، به آن Batch Gradient Descent هم گفته میشود.

تمرین اول : با توجه به فرمول آپدیت زیر، Gradient Descent را کامل کنید.

$$\begin{aligned}W^{[l]} &= W^{[l]} - \alpha dW^{[l]} \\b^{[l]} &= b^{[l]} - \alpha db^{[l]}\end{aligned}$$

مرحله دوم: SGD

نوع دیگری از گرادیان نزولی (SGD) Stochastic Gradient Descent است که برابر است با mini-batch gradient descent که هر mini-batch فقط ۱ مثال دارد.



تمرین دوم: Stochastic Gradient Descent را پیاده سازی کنید.

مرحله سوم: Momentum

از آنجا که mini-batch پس از مشاهده تعدادی نمونه به روز رسانی انجام میشود، شاهد نوسان های زیادی خواهیم بود. برای کاهش این نوسان ها از Momentum استفاده میشود.

Momentum گرادیان های قبلی را نیز مدنظر قرار میدهد و آن را به سمت صاف شدن و نوسان کمتر هدایت میکند.

تمرین سوم: آپدیت پارامتر ها با Momentum را پیاده سازی کنید. (فرمول های زیر را در نظر بگیرید)

$$\begin{cases} v_{dW^{[l]}} = \beta v_{dW^{[l]}} + (1 - \beta) dW^{[l]} \\ W^{[l]} = W^{[l]} - \alpha v_{dW^{[l]}} \end{cases}$$
$$\begin{cases} v_{db^{[l]}} = \beta v_{db^{[l]}} + (1 - \beta) db^{[l]} \\ b^{[l]} = b^{[l]} - \alpha v_{db^{[l]}} \end{cases}$$

مرحله چهارم: ADAM

بهینه ساز ADAM ترکیب RMSProp و Momentum است. قوانین آپدیت آن را در روابط زیر آمده است:

$$\begin{cases} v_{W^{[l]}} = \beta_1 v_{W^{[l]}} + (1 - \beta_1) \frac{\partial J}{\partial W^{[l]}} \\ v_{W^{[l]}}^{corrected} = \frac{v_{W^{[l]}}}{1 - (\beta_1)^t} \\ s_{W^{[l]}} = \beta_2 s_{W^{[l]}} + (1 - \beta_2) \left(\frac{\partial J}{\partial W^{[l]}} \right)^2 \\ s_{W^{[l]}}^{corrected} = \frac{s_{W^{[l]}}}{1 - (\beta_2)^t} \\ W^{[l]} = W^{[l]} - \alpha \frac{v_{W^{[l]}}^{corrected}}{\sqrt{s_{W^{[l]}}^{corrected} + \epsilon}} \end{cases}$$

تمرین چهارم: با توجه به قوانین گفته شده، آپدیت ADAM را پیاده سازی کنید.

مرحله پنجم: بررسی عملکرد بهینه ساز ها روی مدل شبکه عصبی سه لایه

پس از بارگذاری دیتاست، بهینه ساز هایی که پیاده سازی کردید روی شبکه عصبی اجرا کنید و نتایج عملکرد آن ها را مشاهده کنید.