

به نام خدا

گزارش تمرین اول درس یادگیری عمیق

استاد درس: دکتر میرروشندل

علی افشار دگرسی ۱۴۰۴۱۲۲۶۱۵۱۰۱

در ابتدا کتابخانه های مجاز برای استفاده را import می کنیم، به کمک fetch_openml دیتاست mnist را از سایت openml میخوانیم، سپس به این علت که ویژگی های ما بین اعداد ۰ تا ۲۵۵ مقدار گرفته اند، آنها را بر ۲۵۵ تقسیم کرده تا مقادیری بین ۰ و ۱ بگیرند، به اصطلاح آنها را normalize می کنیم تا فرایند آموزش بهتر صورت بگیرد.

سپس به این علت که labels که همان خروجی میباشد، از نوع nomial است، آن را به عدد صحیح cast می کنیم. در خروجی باکس اول میبینیم که تعداد کل نمونه ها ۷۰۰۰۰ عدد می باشد.

سپس ۶۰۰۰۰ نمونه را برای آموزش و ۱۰۰۰۰ نمونه را برای تست جدا می کنیم. حال کلاس ۳ و ۷ را انتخاب کرده و آنها از باقی کلاس ها جدا میکنیم تا مدل را فقط روی دو کلاس آموزش دهیم.

در ادامه کلاس ۳ را به عدد ۰ و کلاس ۷ را به عدد ۱ نگاشت کرده، تا در هنگام آموزش مدل، تابع هزینه که cross entropy می باشد به درستی عمل کند.

سپس به ورودی های مدل که همان x هستند، یک ستون جدید با مقادیر یک اضافه میکنیم تا در فرایند آموزش مدل، وقتی در ماتریس w ضرب میشود، w_0 ، عرض از مبدا یا همان بایاس مدل را در نظر بگیرد.

سپس یک سری توابع کمکی برای فرایند آموزش تعریف می‌کنیم، نکته ای که وجود دارد، با توجه به اینکه تابع هزینه ما cross-entropy می‌باشد، پس convex است، و فقط دارای یک global minimum می‌باشد، پس نیازی به SGD برای فرار از local minimum یا saddle point ها نداریم، و میتوانیم به صورت ساده از Bach Gradient Descent استفاده کنیم که سرعت بیشتری در آموزش به نسبت SGD دارد.

حال یک تابع برای پیاده سازی منطق اصلی کاهش گرادیان تعریف کرده که علاوه بر وزن بهینه، تاریخچه وزن ها، مقادیر خطا و دقت مدل را هم برمی گرداند، در ابتدای این تابع، وزن ها مقدار دهی اولیه و برابر صفر قرار داده می‌شوند، سپس به کمک یک حلقه، برابر تعداد step های تعریف شده، گرادیان را حساب کرده، وزن ها را بروز کرده، و سپس مقدار خطا یا loss را محاسبه کرده و این مقادیر را در حافظه ذخیره میکنیم، سپس از آنجا که تابع loss از نوع convex است، آموزش را تا جایی ادامه می‌دهیم که validation loss از یک مقداری کوچکی مثل 0.0004 بعد از چندین epoch که اینجا ۱۰ در نظر گرفته شده، بهتر نشود، سپس آموزش را متوقف کرده و، وزن بهینه، تاریخچه هزینه و دقت مدل روی داده های train و validation را برمی گردانیم.

سپس در باکس بعدی این تابع را با LR برابر 0.1، تعداد قدم های 2000 و تقسیم داده های train به دو بخش (۱۰۰۰۰ تا اول برای train و باقی برای validation که حدود ۲۰۰۰ عدد است) اجرا میکنیم، در خروجی می‌بینم که بعد از ۹۳۴ epoch، خطا یا loss روی validation کاهش نیافته، پس آموزش متوقف شده است.

سپس در باکس بعدی، نمودار خطا و loss را برای train و validation نمایش داده ایم، میبینم که از یک جایی به بعد خطای validation تغییر نمی‌کند. سپس در باکس بعدی معیار های ارزیابی accuracy، precision، recall، f1-score و confusion matrix را پیاده سازی کرده، و روی داده های تستی که مدل ما تا به حال ندیده بود بررسی می‌کنیم.

در خروجی نهایی میبینیم که دقت بسیار قابل قبولی به دست آمده است:

