

به نام خدا

گزارش تمرین اول درس یادگیری عمیق

استاد درس: دکتر میرروشنده

علی افشار دکرسی ۱۴۰۴۱۲۲۶۱۵۱۰۱

در ابتدا کتابخانه های مجاز برای استفاده را import می کنیم، به کمک fetch_openml دستی است mnist را از سایت openml می خوانیم، سپس به این علت که ویژگی های ما بین اعداد ۰ تا ۲۵۵ مقدار گفته 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 همان خروجی میباشد، از نوع nominal است، آن را به عدد صحیح cast می کنیم. در خروجی باکس اول میبینیم که تعداد کل نمونه ها ۷۰۰۰۰ عدد می باشد. سپس ۶۰۰۰ نمونه را برای آموزش و ۱۰۰۰ 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 می شود، 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 یا 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 وزن‌ها، مقادیر خطای دقت مدل را هم بر می‌گرداند، در ابتدای این تابع، وزن‌ها مقدار دهی اولیه و برابر صفر قرار داده می‌شوند، سپس به کمک یک حلقه، برابر تعداد step‌های تعریف شده، گرادیان را حساب کرده، وزن‌ها را بروز کرده، و سپس مقدار خطای 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 ۰.۰۰۰۴ 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 داده‌های train و validation را بر می‌گردانیم.

سپس در باکس بعدی این تابع را با LR برابر ۰.۱، تعداد قدم‌های ۲۰۰۰ و تقسیم داده‌های train به دو بخش (۱۰۰۰ تا اول برای train و باقی برای validation که حدود ۲۰۰۰ عدد است) اجرا می‌کنیم، در خروجی می‌بینیم که بعد از epoch ۹۳۴، خطای loss را روی validation کاهش نیافته، پس آموزش متوقف شده است.

سپس در باکس بعدی، نمودار خطای loss را برای train و validation نمایش داده ایم، میبینیم که از یک جایی به بعد خطای validation تغییر نمی‌کند. سپس در باکس بعدی معیارهای ارزیابی confusion matrix و accuracy، precision، recall، f1-score داده‌های تستی که مدل ما تا به حال ندیده بود بررسی می‌کنیم.

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

