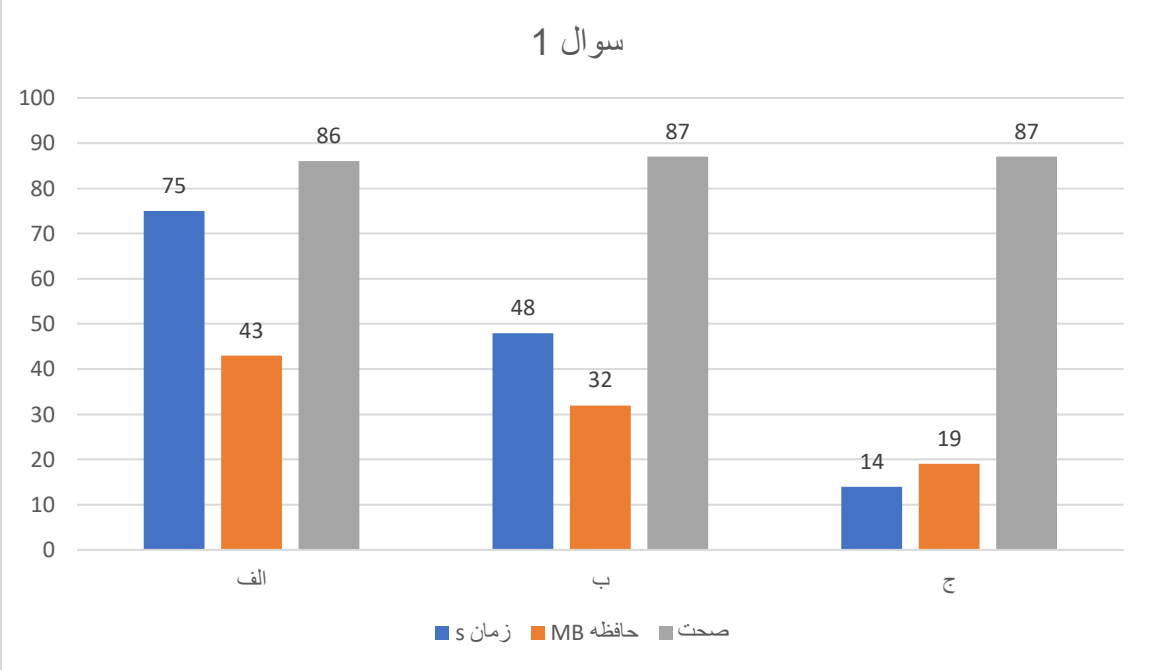
به نام خدا

سامانه های یادگیری ماشین توزیع شده

تمرین کامپیوتری ۲

ابوالفضل اسلامی ۸۱۰۱۹۹۳۷۴

.1

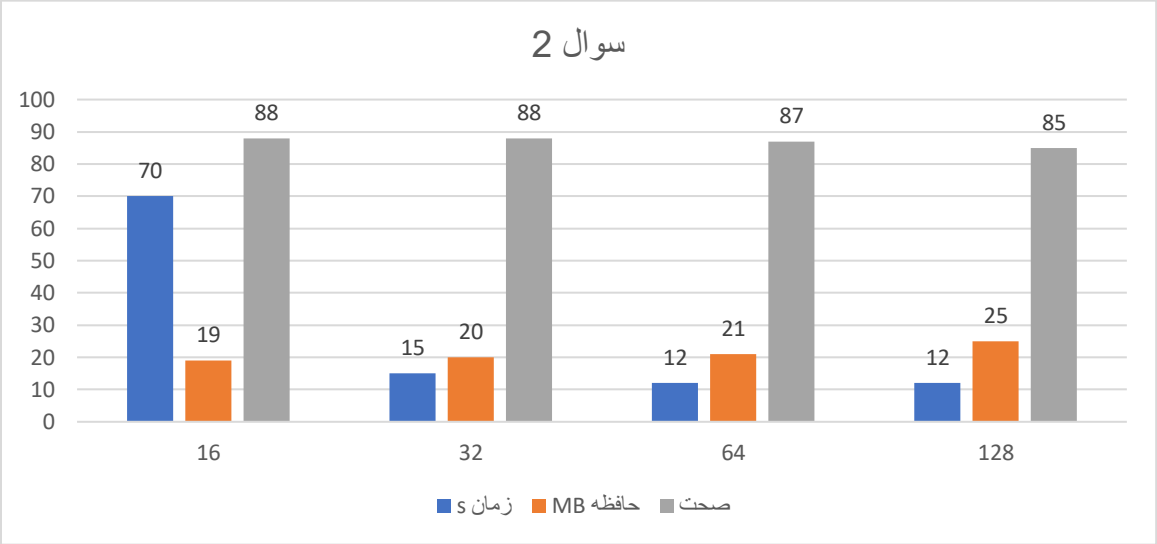


در حالت درستی مدل ها با هم برابر و در حدود 87 درصد است که نشان میدهد موازی سازی تاثیری بر منطق تمرین دادن مدل ندارد.

در حالت الف حافظه و زمان بیشتری از دوحالت دیگر و در حالت ب حافظه و زمان بیشتری از حالت ج مصرف شده است. در حالت الف به دلیل استفاده از یک کارت گرفیک در نگاه اول انتظار میرود که دوبرابر حالت ب منابع مصرف کند که اینگونه نیست. یکی از دلایل این موضوع میتواند سربار انتقال داده به کارت گرافیک باشد.

حالت ج از بقیه حالت ها سریع تر است و حافظه کمتری مصرف میکند که به دلیل پیکربندی است که torchrun انجام میدهد.

.2



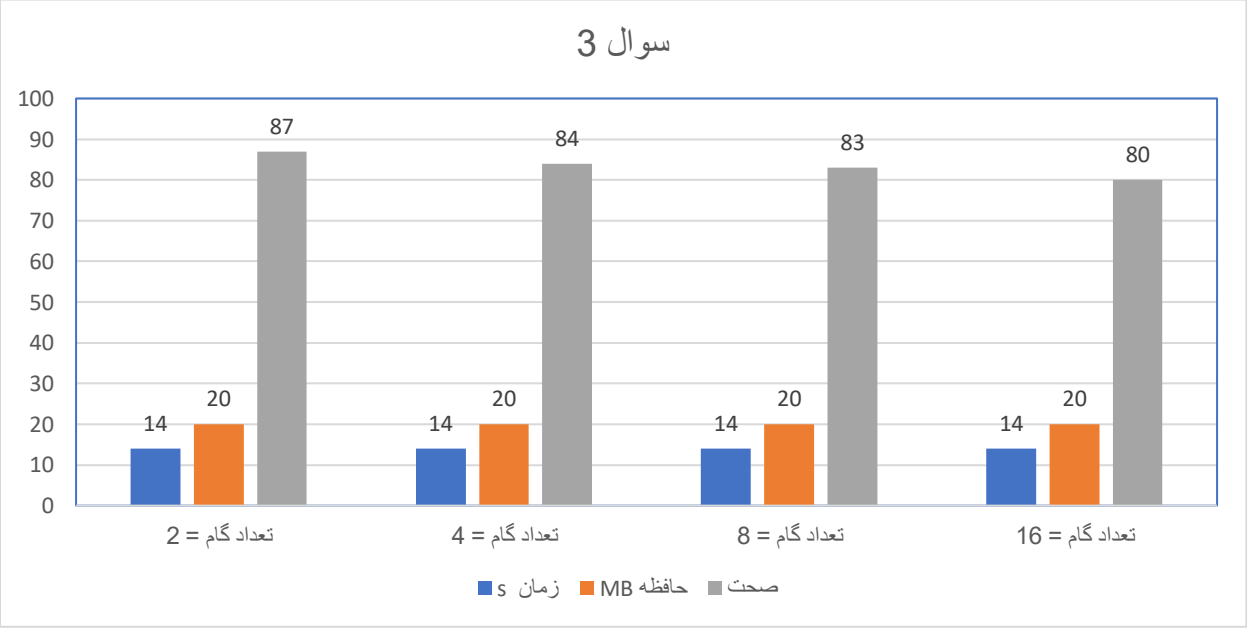
هر چه سایز batch بیشتری میشود زمان اجرا کاهش میباید , دقت مدل کاهش و حافظه مصرفی افزایش مییابد.

مشخصا سایز بالاتر batch باعث استفاده بیشتر از حافظه میشود.

افزایش اندازه دسته ممکن است مدل به دادههای آموزشی بیش از حد برازش (overfit) کند و برای دادههای جدید کاهش دقت داشته باشد.

رفتاری که مشاهده میکنیم، که زمان اجرای دسته 16 به مراتب بیشتر از 32 طول میکشد، اما 64 و 128 نزدیک 32 هستند، ممکن است تحت تأثیر مستقیم معماری خاص شبکه عصبی و GPU نحوه بهرهمندی آن از قابلیتهای پردازش موازی باشد.

.3



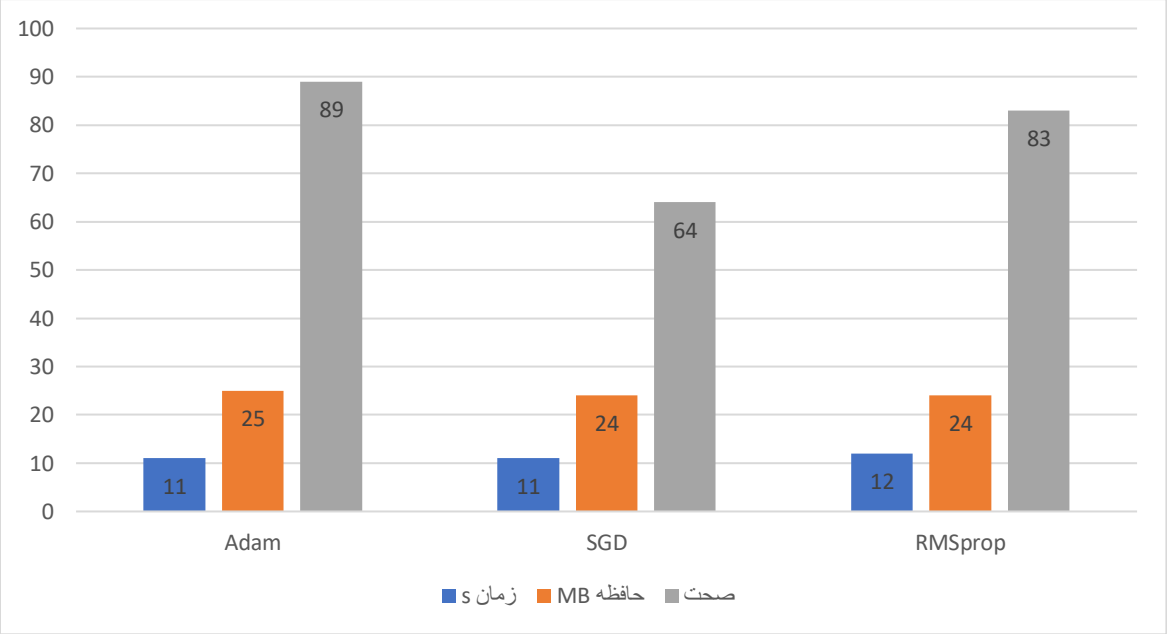
تکنیک Accumulated Gradient یا همان تجمیع گرادیان، یک روش در آموزش شبکه‌های عصبی است که این امکان را می‌دهد که گرادیان‌های محاسبه شده در چندین مینی‌بچ (batch) را جمع زده و بعداً از این جمع‌آوری را برای به‌روزرسانی وزن‌ها استفاده شود. این روش معمولاً در مواردی مورد استفاده قرار می‌گیرد که محدودیت‌های حافظه یا منابع سخت‌افزاری باعث می‌شود که اندازه دسته بزرگتری را نتوان استفاده کرد.

افزایش تعداد گام‌های تجمیع گرادیان ممکن است منجر به نیاز به حافظه بیشتر شود، زیرا گرادیان‌های محاسبه شده برای هر مینی‌بچ در حافظه نگهداری می‌شوند تا زمانی که تعداد تعیین شده از گام‌ها جمع‌آوری شوند. این موضوع می‌تواند بهبود بخشیدن به مدیریت محدودیت‌های حافظه کمک کند.

افزایش تعداد گام‌های تجمیع گرادیان معمولاً منجر به زمان آموزش کمتری می‌شود. زیرا بروزرسانی وزن‌ها فقط پس از تجمیع گرادیان‌ها انجام می‌شود و هر بار محاسبه گرادیان از یک مینی‌بچ صورت می‌گیرد. این امکان را فراهم می‌کند که از اندازه دسته بزرگتری استفاده کنید و بهبود بهره‌وری آموزش داشته باشید.

افزایش تعداد گام‌های تجمیع ممکن است باعث کاهش دقت شود، زیرا اطلاعات جزئی‌تری از داده را به مدل ارائه می‌دهد.

.4



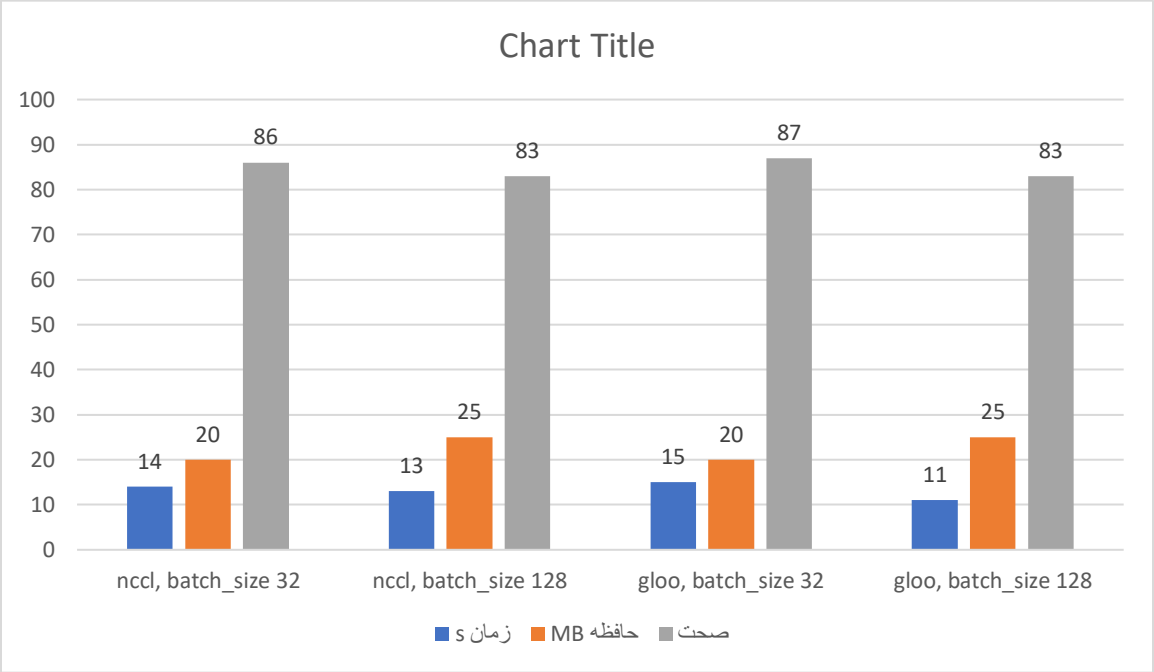
به نظر می‌رسد کهOptimizer های Adam و SGD زمان آموزش مشابهی دارند (حدود 11 ثانیه)، در حالی که RMSprop کمی بیشتر است (12 ثانیه). این نتایج ممکن است به معنای این باشد که Adam و SGD در این مورد موازنه بهتری بین سرعت آموزش و دقت مدل دارند.

Optimizer Adamبا حافظه مصرفی 25 مگابایت بیشترین مقدار حافظه را مصرف کرده است. به نظر می‌رسد که حافظه مصرفی SGDو RMSprop کمی کمتر است (24 مگابایت). این نتایج نشان‌دهنده این موضوع است که Adam ممکن است برای مسائل با حافظه محدود، یک انتخاب کمتر مناسب باشد.

در مورد دقت مدل، Adamبا دقت 89 درصد بالاترین عملکرد را دارد. این می‌تواند نشان‌دهنده این باشد که Adam برای مسائل خاصی (حداقل در مورد دقت) ممکن است بهترین عملکرد را داشته باشد. و SGD دقت بسیار پایین تری نسبت به بقیه دارد.

در مجموع Adam در این مثال از بقیه بهتر عمل میکند.

.5



در سه فاکتور مورد نظر عملکرد هر دو پروتکل تقریبا مشابه است و در این مثال نمیتوان مقایسه دقیقی انجام داد.