شبیه‌سازی و ارزیابی سیاست‌های زمان‌بندی خوشه‌ای آگاه از ناهمگونی برای بارهای کاری یادگیری عمیق

دانشجو: حامد حسین پور

استاد راهنما: دکتر احمد خونساری

دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران

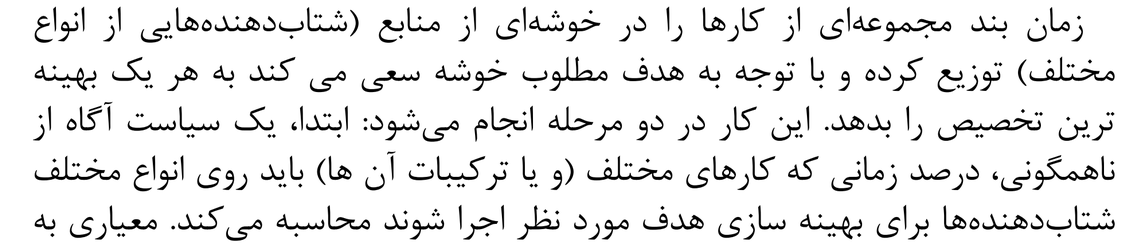
خلاصه

شتاب دهندهای متنوعی برای آموزش مدل های یادگیری عمیق استفاده شده اند که هر کدام در برابر هر مدل، رفتار ناهمگونی نشان می دهند. زمان بندهای موجود برای خوشه هایی از شتاب دهنده ها که برای توزیع منابع بین چندین کاربر استفاده می شوند، روش های مختلفی را برای بهینه سازی اهدافشان به کار برده اند؛ اما اکثراً این ناهمگونی را در نظر نمی گیرند.

زمان بند معرفی و شبیه سازی شده، یک زمان بند ناهمگونی–آگاه است که طیف وسیعی از سیاست های زمان بندی را در کنار هم در نظر گرفته و هر یک را به عنوان مسئله بهینه سازی مطرح می کند و با استفاده از یک معیار که به آن گذردهی موثر می گوییم، آن ها را به نسخه آگاه از ناهمگونی خود تبدیل می کند. سپس با استفاده از یک سازوکار گردشی و بر اساس سیاست مدنظر، از ایده آل بودن اختصاص صورت گرفته به شتاب دهنده ها و اجرا شدن آن مطمئن می شود.

ساختار و روش

زمان بند مجموعه‌ای از کارها را در خوشه‌ای از منابع (شتاب‌دهنده‌هایی از انواع مختلف) توزیع کرده و با توجه به هدف مطلوب خوشه سعی می کند به هر یک بهینه ترین تخصیص را بدهد. این کار در دو مرحله انجام می‌شود: ابتدا، یک سیاست آگاه از ناهمگونی، درصد زمانی که کارهای مختلف (و یا ترکیبات آن ها) باید روی انواع مختلف شتاب‌دهنده‌ها برای بهینه سازی هدف مورد نظر اجرا شوند محاسبه می‌کند. معیاری به نام گذردهی برای هر کار روی هر نوع شتاب‌دهنده که می‌تواند توسط کاربر ارائه شود یا توسط تخمین گر گذردهی به صورت پویا اندازه‌گیری شود، به عنوان ورودی و متناسب با هر سیاست داده می شود و به کمک آن ماتریس تخصیص محاسبه می گردد. تخصیص‌های داده شده فقط باید بین بازمحاسبه های آن ها در هر دوره رعایت شوند و زمان بند می تواند سیاست خود را هنگام وقوع یک رویداد بازنشانی یا در فواصل زمانی دوره‌ای، دوباره محاسبه کند. با توجه به خروجی تخصیص مدنظر سیاست، یک ساز و کار زمان‌بندی گردشی به هر کار زمان برای اجرا روی منابع مختلف را اعطا کرده و بین منابع جابجا می‌کند تا از درستی درصد زمانی صرف شده هر کار روی منابع مطمئن شود و کاملا شبیه تخصیص بهینه‌ای باشد که از قبل مشخص شده است. روند کلی کار در شکل قابل مشاهده می باشد.

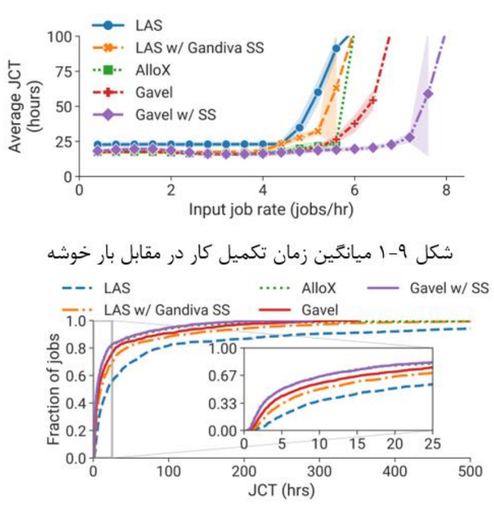


کارها در چارچوب‌هایی مانند PyTorch یا TensorFlow نوشته می‌شوند. تخمین‌گر گذردهی، در صورت نیاز اندازه‌گیری‌های عملکرد را برای هر کار قابل اجرا بر روی هر نوع شتاب‌دهنده در دسترس به دست می‌آورد؛ سپس ماتریس تخصیصی را محاسبه می‌کند که یک هدف و سیاست مشخص شده توسط کاربر و مانند عدالت را به صورت بهینه اجرا می‌کند. سازوکار زمان‌بندی گردشی این تخصیص محاسبه شده را به عنوان ورودی می‌پذیرد و تصمیم می گیرد کارها در هر دور در چه نسبت‌هایی روی منابع قرارگیرند و به طور کامل تخصیص محاسبه شده را اعمال می‌کند.

نتایج

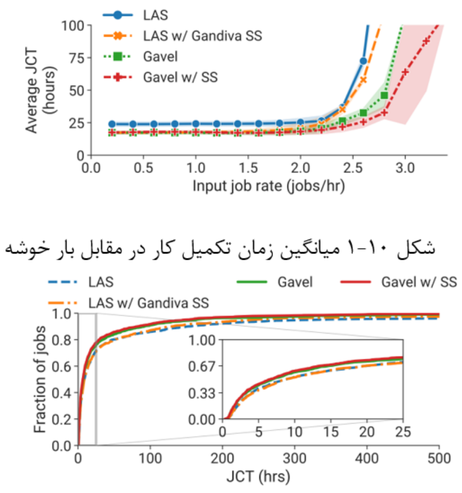
سیاست‌های آگاه از ناهمگونی ما، اهداف نهایی را روی یک خوشه شبیه‌سازی شده بهبود می‌بخشند و این امکان را فراهم می‌کند که از نرخ ورودی متوسط بالاتری برای کارها یادگیری پشتیبانی کند، در حالی که اهدافی مانند میانگین زمان اتمام کار را بین ۱/۴ تا ۳/۵ برابر، زمان پایان را تا ۲/۵ برابر و هزینه را تا ۱/۴ برابر در مقایسه با سیاست های ناآگاه از ناهمگونی بهبود می‌بخشد.

ما بر این باوریم که تاخیرهای بالای ۳۰ دقیقه برای خوشه‌های بزرگ هنوز بهتر از زمان‌بندهای غیرقطعی هستند که در آن‌ها کارها تاخیرهای طولانی در صف را تجربه می‌کنند یا زمان‌بندهای قطعی با سیاست‌های بی‌توجه به ناهمگون که منجر به بدتر شدن مقادیر هدف می‌شوند.



شکل ۹-۱ میانگین زمان تکمیل کار در مقابل بار خوشه

شکل ۹-۲ CDF زمان های تکمیل کار (نرخ کار ورودی = ۵.۶ شغل در ساعت)



شکل ۱۰-۱ میانگین زمان تکمیل کار در مقابل بار خوشه

شکل ۱۰-۲ CDF زمان های تکمیل کار (نرخ کار ورودی = ۵.۶ شغل در ساعت)

جمع بندی

اگرچه ما در این پژوهش روی تخصیص منابع ناهمگون برای بارهای کاری آموزش DNN تمرکز کردیم، معتقدیم که زمان بند می‌تواند برای بارهای کاری غیرDNN نیز استفاده شود. بارهای کاری دیگری که قابل اجرا بر روی GPU هستند، مانند شبیه‌سازی‌ها، می‌توانند در نظر گرفته شوند. همچنین معتقدیم که بینش فنی اصلی ارائه شده در این پروژه که فرمول‌بندی سیاست‌های متنوع زمان‌بندی به عنوان مسائل بهینه‌سازی است، به طور گسترده‌ای قابل اجرا است و می‌توان از آن برای استقرار آسان‌تر سیاست‌ها در خوشه‌های یادگیری عمیق همگون و همچنین در خوشه‌های CPU نیز استفاده کرد.

مراجع اصلی

1. **Narayanan, D., Santhanam, K., Kazhamiaka, F., Phanishayee, A., Zaharia, M. (2020). Heterogeneity-Aware Cluster Scheduling Policies for Deep Learning Workloads. Usenix.**
2. **Gao, W., Hu, Q., Ye, Z., Sun, P., Wang, X., Luo, Y., Zhang, T., Wen, Y. (2022). Deep Learning Workload Scheduling in GPU Datacenters: Taxonomy, Challenges and Vision. arXiv.**
3. **Candes, E., J., Plan, Y. (2010). Matrix Completion with Noise. Proceedings of the IEEE, 98(6):925–936, 2010.**