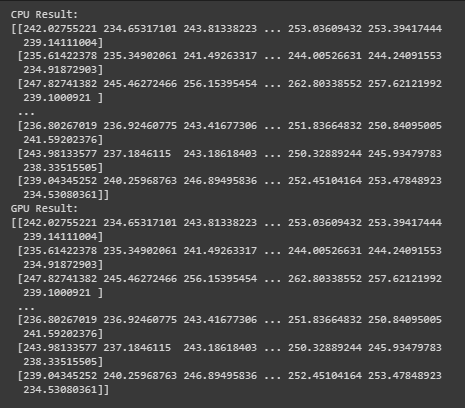
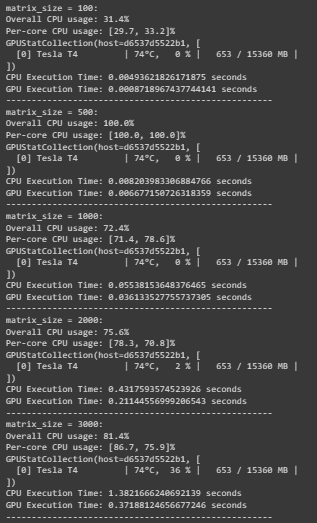
این الگوریتم برای ضرب دو ماتریس با استفاده از روش‌های CPU و GPU پیاده‌سازی شده است. الگوریتم به شرح زیر عمل می‌کند:

1. در ابتدا، اندازه ماتریس (matrix\_size) را تعیین می‌کنیم و دو ماتریس مربوطه matrix\_a) و (matrix\_b را به اندازه matrix\_size × matrix\_size با استفاده از تابع np.random.rand ایجاد می‌کنیم. این تابع مقادیر تصادفی بین 0 و 1 را در ماتریس تولید می‌کند.
2. تابع cpu\_matrix\_multiplication به عنوان الگوریتم ضرب ماتریس با استفاده از CPU پیاده‌سازی شده است. در این تابع، زمان شروع اجرای الگوریتم (start\_time) را ثبت می‌کنیم.
   * ماتریس‌های ورودی matrix\_a) و (matrix\_b را با استفاده از تابع np.dot ضرب می‌کنیم و نتیجه را در متغیر cpu\_result ذخیره می‌کنیم.
   * زمان پایان اجرای الگوریتم (end\_time) را ثبت می‌کنیم و زمان اجرا را با اختلاف end\_time و start\_time محاسبه می‌کنیم.
   * نتیجه ضرب ماتریس (cpu\_result) و زمان اجرا (execution\_time) را برمی‌گردانیم.
3. تابع gpu\_matrix\_multiplication نیز همان عملکرد ضرب ماتریس را با استفاده از GPU پیاده‌سازی می‌کند. در این تابع:
   * ماتریس‌های ورودی matrix\_a) و (matrix\_b را به فرمت قابل قبول برای GPU تبدیل می‌کنیم و در متغیرهای gpu\_matrix\_a و gpu\_matrix\_b ذخیره می‌کنیم.
   * با استفاده از تابع cp.dot، ضرب ماتریس را با استفاده از GPU انجام می‌دهیم و نتیجه را در متغیر gpu\_result ذخیره می‌کنیم.
   * سپس نتیجه را به فرمت قابل قبول برای CPU تبدیل کرده و در متغیر cpu\_result ذخیره می‌کنیم.
   * زمان شروع و پایان اجرای الگوریتم را ثبت کرده و زمان اجرا را محاسبه می‌کنیم.
   * نتیجه ضرب ماتریس (cpu\_result) و زمان اجرا (execution\_time) را برمی‌گردانیم.
4. در حلقه for با تعداد تکرار 5 بار، مراحل زیر تکرار می‌شوند:
   * در هر مرحله، شماره مرحله (i+1) را نمایش می‌دهیم.
   * با استفاده از تابع psutil.cpu\_percent، درصد استفاده کلی از CPU را در متغیر cpu\_percent ذخیره می‌کنیم.
   * با استفاده از تابع psutil.cpu\_percent با پارامتر percpu=True ، درصد استفاده از هر هسته از CPU را در لیست cpu\_percent\_per\_core ذخیره می‌کنیم.
   * مشخصات مربوط به استفاده از پردازنده گرافیکی را با استفاده از تابع gpustat.new\_query نمایش می‌دهیم.
   * ابتدا الگوریتم ضرب ماتریس با استفاده از روش CPU اجرا می‌شود و نتیجه ضرب و زمان اجرا را در متغیرهای cpu\_result و cpu\_execution\_time ذخیره می‌کنیم.
   * سپس الگوریتم ضرب ماتریس با استفاده از روش GPU اجرا می‌شود و نتیجه ضرب و زمان اجرا را در متغیرهای gpu\_result و gpu\_execution\_time ذخیره می‌کنیم.
   * در صورت نیاز، نتایج حاصل از ضرب ماتریس با روش‌های CPU و GPU را نمایش می‌دهیم.
   * در نهایت، زمان اجرا برای هر دو روش را نمایش می‌دهیم.

این کد برای مقایسه زمان اجرا و استفاده از منابع سخت‌افزاری بین روش‌های CPU و GPU در عملیات ضرب ماتریس مورد استفاده قرار می‌گیرد.

با توجه به ، می‌توانیم تحلیلی بر روی آن‌ها انجام دهیم:

1. مشخصات استفاده از پردازنده مرکزی (CPU) و پردازشگر گرافیکی (GPU) در هر مرحله نمایش داده می‌شود. می‌توان توجه کرد که مقدار استفاده از پردازنده مرکزی در هر مرحله متفاوت است و ممکن است به دلایل مختلفی مانند بارکاری سیستم تغییر کند.
2. نتیجه ضرب ماتریس برای روش‌های CPU و GPU به صورت یکسان در همه مراحل نمایش داده می‌شود. این نتایج باید یکسان باشند زیرا عمل ضرب ماتریس با استفاده از هر دو روش صورت می‌گیرد و باید به نتایج یکسانی منجر شود.
3. زمان اجرای الگوریتم برای هر دو روش CPU) و (GPU نمایش داده می‌شود. می‌توان توجه کرد که در هر مرحله، زمان اجرای الگوریتم برای روش GPU کمتر از زمان اجرای الگوریتم برای روش CPU است. این نشان می‌دهد که استفاده از پردازشگر گرافیکی (GPU) بهبود قابل توجهی در سرعت اجرای عملیات ضرب ماتریس دارد.
4. استفاده از پردازنده مرکزی (CPU) در همه مراحل بالا استفاده بیشتری نسبت به پردازشگر گرافیکی (GPU) دارد. این ممکن است به دلیل استفاده‌های دیگر از پردازنده مرکزی در سیستم باشد، اما ممکن است نیاز به بهینه‌سازی بیشتر در استفاده از GPU و استفاده بهینه‌تر از قابلیت‌های آن باشد.
5. مشاهده می‌شود که در مرحله 3، مقدار استفاده از پردازنده مرکزی (CPU) نسبت به مراحل قبلی بیشتر است و همچنین زمان اجرای الگوریتم برای هر دو روش CPU) و (GPU نسبت به مراحل قبلی افزایش یافته است. این می‌تواند به دلیل بارکاری سیستم در آن مرحله باشد و نشان می‌دهد که می‌تواند باعث افزایش زمان اجرای الگوریتم باشد.

با توجه به تحلیل فوق، می‌توان نتیجه گرفت که استفاده از پردازشگر گرافیکی (GPU) در عملیات ضرب ماتریس می‌تواند زمان اجرا را بهبود بخشیده و سرعت اجرای الگوریتم را افزایش دهد. همچنین، بارکاری سیستم و استفاده از منابع سخت‌افزاری (مانند پردازنده مرکزی) می‌تواند تأثیر قابل توجهی بر زمان اجرا داشته باشد و باید در بررسی‌های بعدی در نظر گرفته شود.

