بخش ۱

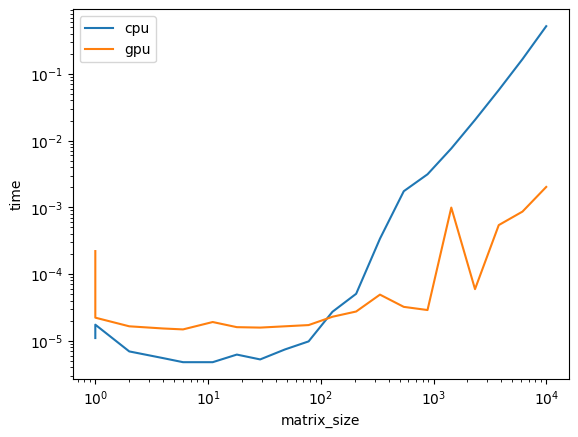
به دلیل اینکه سیستم شخصی GPU نداشت از colab با GPU استفاده شده است.

بخش ۳-۱

با استفاده از کتابخانه PyTorch و Numpy ما به اعمال محاسبات ماتریسی پرداختیم

ابتدا یک لیست از سایز ماتریس ها در فضای لگاریتمی تهیه شد و سپس به ازای هر سایز ماتریس ۲ عدد ماتریس ۲ بعدی برای اعمال ضرب درایه ای به صورت تصادفی ایجاد شد و ضرب درایه ای با استفاده از نامپای بر روی CPU و با استفاده از PyTorch بر روی GPU(CUDA) اعمال گردید.

نمودار زیر نمایش دهنده زمان بر حسب ابعاد ماتریس ها می باشد.

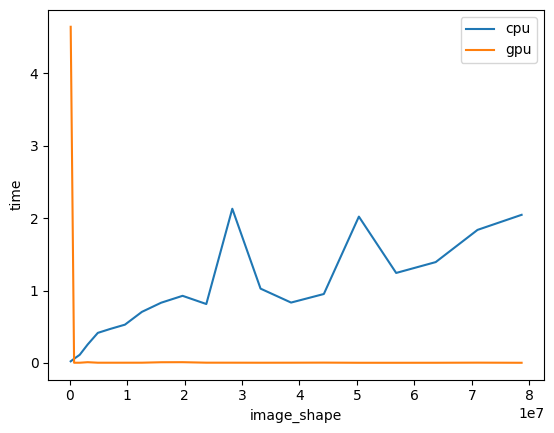


بخش ۳-۲

در این بخش ابتدا تصویر معروف لنا را دانلود کرده و می خوانیم و سپس با استفاده از PyTorch فیلتر گاوسی را برای بلور کردن تصویر بر روی آن اعمال می کنیم.

تصویر را در مقیاس های مختلف به شبکه کانولوشنی داده و با استفاده از تابع to نوع CPU یا GPU(CUDA) را انتخاب میکنیم

نمودار زیر نمایش دهنده زمان بر حسب سایز عکس (حاصل ضرب ابعاد ماتریس ۳بعدی عکس) می باشد:



در هر دو نمودار مشاهده می شود تایم CPU با افزایش ابعاد عکس و ماتریس های ورودی افزایش پیدا می کند در حالی که تایم GPU تغییرات خیلی کمی دارد. پیک اول GPU. به دلیل آماده سازی اولیه GPU و اختصاص فضای RAM می باشد.

در ابعاد پایین CPU بهتر عمل می کند چرا که انتقال داده ها ازRAM به GRAM و بازگردانی آن به RAM مقداری زمان می برد که این زمان بیشتر از زمان محاسبه ماتریس توسط CPU می باشد. لذا در ابعاد پایین این مقدار زمان مهم می باشد.

در محاسباتی که سلسله مراتبی هستن و برای محاسبات بعدی نیاز به نتیجه مرحله قبل می باشد استفاده از CPU بهتر است چراکه GPU برای موازی سازی محاسبات طراحی شده است. شبکه های عصبی بازگشتی مانند RNN ها و LSTM از این دست شبکه ها می باشند.