



تمرین برنامه نویسی شماره 1



عنوان: برنامه نویسی موازی با استفاده از موازی سازی در سطح داده

درس: برنامه نویسی موازی

استاد درس: دکتر سعید صفری

طراحان: [امیرحسین راحتی](#)، [هاتف رضائی](#)

نیمسال اول سال تحصیلی 1404-05

هدف

در این تمرین می خواهیم با استفاده از دستورات و روش SIMD تعدادی الگوریتم ساده را پیاده سازی و میزان افزایش کارایی را حساب کنیم. برای هر قسمت نسخه سریال و نسخه ی معادل موازی آن را پیاده سازی و مقایسه می کنیم. توجه داشته باشید که برای خواندن و نمایش عکس ها می توانید از OpenCV استفاده کنید.

قسمت اول

محاسبه واریانس

در این بخش باید واریانس یک مجموعه داده ی عددی را با استفاده از دستورات SIMD به صورت موازی محاسبه کنید. واریانس شاخصی از میزان پراکندگی داده ها است و با رابطه ی زیر تعریف می شود:

$$Var(X) = \frac{1}{N} \sum_{i=1}^N (\bar{x} - x_i)^2$$

ابتدا باید یک آرایه از اعداد اعشاری با اندازه ی دلخواه به صورت تصادفی تولید کنید. روی داده های یکسان دو نسخه سریال و موازی برنامه را پیاده سازی کنید و میزان Speedup بدست آمده را گزارش کنید.

قسمت دوم

اضافه کردن ته نقش¹ به تصویر

در این بخش هدف آن است که یک تصویر ته نقش را به صورت تدریجی بر روی تصویر اصلی قرار دهیم. شدت شفافیت² ته نقش در سراسر تصویر یکنواخت نیست، بلکه از گوشه ی بالا-چپ تا گوشه ی پایین-راست به صورت خطی افزایش می یابد. به عبارت دیگر، در پیکسل های بالا-چپ کم رنگ تر است و در پیکسل های پایین-راست به تدریج پررنگ تر دیده می شود. ضریب شفافیت هر پیکسل (α) با استفاده از رابطه ی زیر محاسبه می شود:

$$\alpha(x, y) = \frac{x + y}{w + h}$$

¹ Watermark

² Transparency

که در آن w و h به ترتیب عرض و ارتفاع تصویر اصلی هستند. در هر پیکسل از تصویر خروجی، مقدار نهایی هر کانال رنگ (R,G,B) باید طبق رابطه‌ی زیر محاسبه شود:

$$Out = \alpha(x, y) \times Watermark + (1 - \alpha(x, y)) \times Base$$

تصاویر مورد نیاز برای این بخش در صفحه درس بارگذاری شده است.

قسمت سوم

پیدا کردن لبه‌ها³

الگوریتم Sobel

فیلترهای sobel یکی از شناخته شده ترین فیلترها برای تشخیص لبه ها می باشند. از این دو فیلتر یکی از فیلترها برای تشخیص لبه های افقی و یکی برای تشخیص لبه های عمودی عکس می‌باشند. در این بخش باید با استفاده از فیلترهای sobel لبه های تصویر را تشخیص دهیم.

```
sobel_x = [
    [-1,0,1],
    [-2,0,2],
    [-1,0,1]]
sobel_y = [
    [-1,-2,-1],
    [0,0,0],
    [1,2,1]]
gaussian_blur = [
    [1,2,1],
    [2,4,2],
    [1,2,1]]
blurred = convolve(gray, gaussian_blur)
edge_x = convolve(blurred, sobel_x)
edge_y = convolve(blurred, sobel_y)
magnitude = max(sqrt(pow(edge_x,2) + pow(edge_y,2)),255)
```

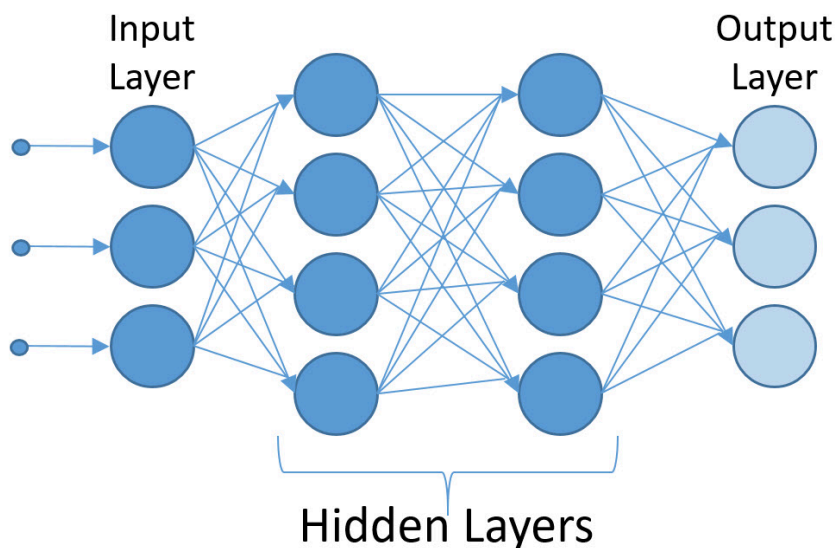
برای تصویر مورد نیاز این بخش از `base.jpg` قسمت قبل استفاده فرمایید.

³ Edge detection

قسمت چهارم

شبکه ی عصبی کاملاً متصل⁴

در شبکه‌های عصبی، هر لایه از تعدادی گره⁵ تشکیل شده است که با گره‌های لایه‌ی قبل و بعد خود ارتباط کامل دارند. به این نوع ساختار، شبکه‌ی عصبی کاملاً متصل گفته می‌شود.



در این شبکه‌ها، هر گره خروجی، حاصل جمع وزن‌دار ورودی‌های مرحله قبل به‌اضافه‌ی یک مقدار بایاس است. سپس یک تابع فعال‌ساز⁶ مانند ReLU یا Sigmoid روی این مقدار اعمال می‌شود. از آن‌جا که این عملیات برای تمام نورون‌های خروجی به‌صورت مستقل تکرار می‌شود، این بخش محاسبات زیادی دارد و اجرای موازی آن‌ها اهمیت بالایی دارد.

در این تمرین هدف آن است که یک شبکه‌ی عصبی کاملاً متصل را با استفاده از دستورات SIMD به‌صورت موازی پیاده‌سازی کنید. ساختار شبکه شامل سه لایه است:

- لایه‌ی ورودی با ۸ گره
- لایه‌ی پنهان با ۱۶ گره
- لایه خروجی با ۸ گره

⁴ Fully connected neural network

⁵ Neuron

⁶ Activation Function

در ابتدای برنامه باید تمام داده‌ها شامل وزن‌ها، بایاس‌ها و ورودی‌ها به صورت تصادفی مقداردهی اولیه شوند. سپس دو نسخه سریال و موازی با داده‌های یکسان محاسبه شوند.

نکات:

- صرفاً نیاز است که محاسبه خروجی هر نورون انجام شود و نیازی به پیاده سازی مراحل training نیست.
- برای Activation function هر تابع دلخواه مورد قبول است.
- برای بهتر مشخص شدن Speedup میتوانید محاسبات را چندین بار تکرار کنید یا اندازه لایه های شبکه را با حفظ نسبت فعلی افزایش دهید.

نکات و نحوه تحویل

- برای باز کردن، ذخیره کردن و اعمال تغییرات روی تصویرها در سوال‌های پردازش تصویر، می‌توانید از OpenCV استفاده کنید.
- برای هر کدام از سوال‌ها نسبت زمان حالت سریال به حالت موازی را در قالب speedup گرفته شده در خروجی چاپ کرده و در گزارش خود توجیه کنید.
- در این تمرین، تنها امکان موازی‌سازی با استفاده از دستورالعمل‌های SIMD را خواهید داشت و استفاده از روش‌های دیگر، نمره‌ای در پی ندارد.
- توجه کنید که گرفتن speedup معقول بخشی از نمره تمرین شما را شامل می‌شود.
- در نظر داشته باشید، نسخه‌ی سریال باید در حالت بهینه بوده و پیاده‌سازی نسخه‌ی موازی با نسخه‌ی سریال منطبق باشد. قسمتی از ارزیابی پیاده‌سازی شما به همین موضوع اختصاص پیدا خواهد کرد.
- کدهای شما باید به زبان C++/C و در سیستم عامل Windows و یا Linux قابل کامپایل و اجرا باشند.
- حتما در ابتدای فایل برنامه ارسالی، نام، نام خانوادگی و شماره‌ی دانشجویی اعضای گروه ذکر شود.
- تنها یکی از اعضای گروه پاسخ تمرین را آپلود کند.
- در محل بارگذاری در صفحه درس، فایل‌ها و کدهای مورد نیاز به همراه گزارش پروژه را بارگذاری نمایید.
- ممکن است به صورت تصادفی پروژه از برخی از شما تحویل (حضور) گرفته شود.
- در صورت داشتن سوال می‌توانید از طریق ایمیل با طراحان تمرین در ارتباط باشید.
- هدف این تمرین یادگیری شماسست، لطفا تمرین را خودتان انجام دهید.
- در صورت محرز شدن تقلب، مطابق با سیاست‌های درس برخورد خواهد شد.