

Ali Alsharif

CSC 671

Measure the speedup of vectorized matrix-matrix multiplication vs. plain python

The purpose of this assignment is to measure the speedup ratio in matrix-matrix multiplication using vectors vs plain Python. It involves generating random matrices, performing multiplication using both plain Python and PyTorch's vectorized operation, measuring the execution time, and plotting a graph to show the speed improvement.

Data generation

We first generate two matrices , W and X, with dimensions (90 x m) and (m x 110), where m takes values from 10 to 100.

Matrix Multiplication Functions

Code (Plain Python):

```
def plain_python_multiply(W, X):  
    result = np.zeros((90, 110))  
    for i in range(90):  
        for j in range(110):  
            for k in range(len(W[0])):  
                result[i][j] += W[i][k] * X[k][j]  
    return result
```

Code (Vectorized):

```
def vectorized_multiply(W_t, X_t):  
    return torch.matmul(W_t, X_t)
```

Description:

We create two matrix multiplication functions. One using plain Python and the other using Pytorch's vectorized operation(matmul).

Execution Time

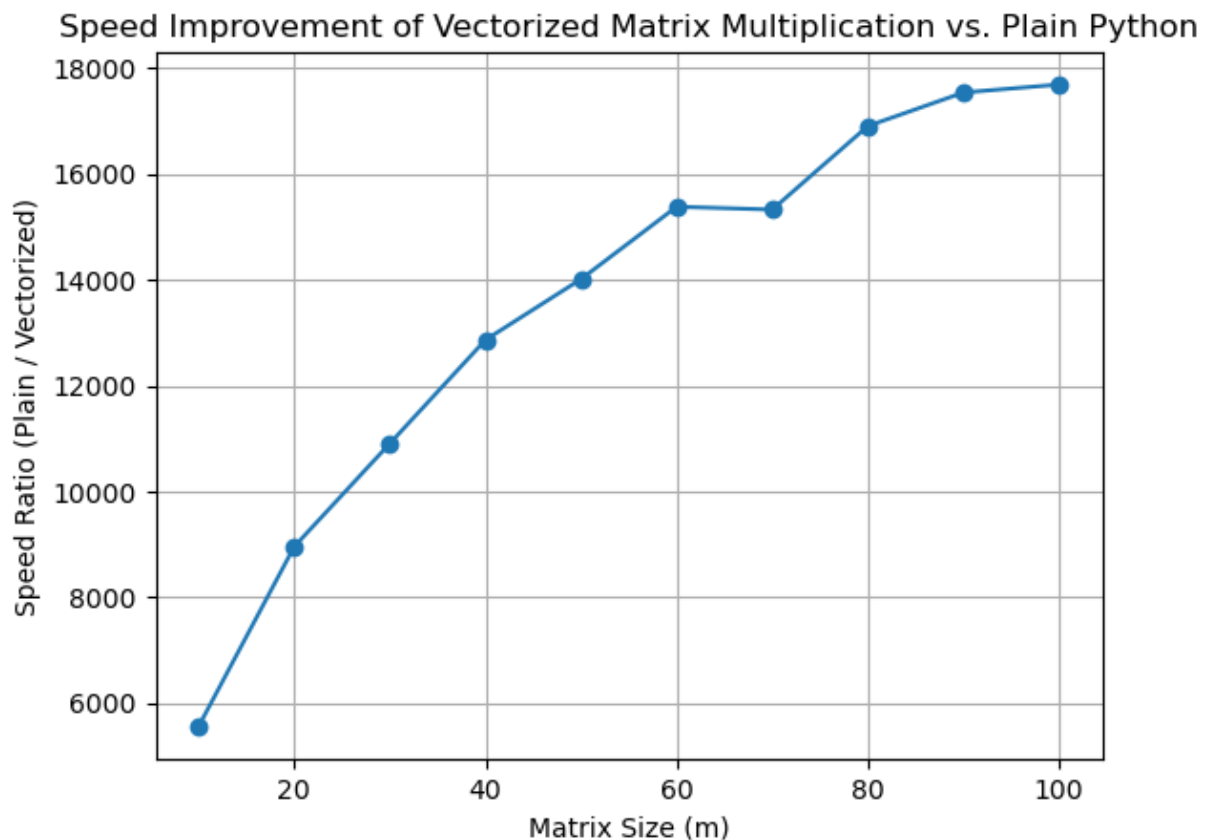
We then use (%timeit -o) to measure the execution time for both functions. it runs with 10 different values of m.

Speed Ratio Calculation

The speed ratio is calculated by dividing the plain Python time over the Vectorized time.

Plotting

We then plot the speedup ratios given to us.



Conclusion

After the graph is plotted, it shows us how the vectorized approach consistently outperforms the plain Python, and that demonstrates how PyTorch is more efficient because of its optimized operations.