akMatMultCombined

Axel Kemper

How to combine matrix multiplication algorithms?

A bilinear non-commutative algorithm to multiply two matrices $[a_{ij}]$ and $[b_{mn}]$ with K elementary products can be written as:

$$[c_{pq}] = \sum_k \gamma_{pqk} \sum_{ijk} \alpha_{ijk} a_{ij} \sum_{mnk} \beta_{mnk} b_{mn}$$

If the algorithm multplies an $M \times N$ matrix with an $N \times P$ matrix, it is denoted as $< M \times N \times P >$.
If two such algorithms with K_x and K_y products respectively are given, they can be combined. This results in an algorithm $< M_x M_y \times N_x N_y \times P_x P_y >$ with $K_{xy} = K_x \times K_y$ elementary products.

Examples

The algorithm $< 2 \times 2 \times 2 >$ with 7 products (Volker Strassen) combined with the algorithm $< 3 \times 3 \times 3 >$ with 23 products (Julian Laderman) results in an algorithm $< 6 \times 6 \times 6 >$ with 161 products.

The algorithm $< 2 \times 3 \times 2 >$ with 11 products combined with the algorithm $< 3 \times 2 \times 3 >$ with 15 products (Hopcroft) results in an algorithm $< 6 \times 6 \times 6 >$ with 165 products.

Notation

The first algorithm (denoted as x) operates with matrices of matrices. The second algorithm (denoted as y) multiplies matrices of scalars as usual. Using capital letters for matrices and lowercase letters for scalars, the two algorithms can be written as:

$$[C_{p_x q_x}] = \sum_{k_x} \gamma^x_{p_x q_x k_x} \sum_{i_x j_x k_x} \alpha^x_{i_x j_x k_x} A_{i_x j_x} \sum_{m_x n_x k_x} \beta^x_{m_x n_x k_x} B_{m_x n_x}$$

$$[c_{p_y q_y}] = \sum_{k_y} \gamma^y_{p_y q_y k_y} \sum_{i_y j_y k_y} \alpha^y_{i_y j_y k_y} a_{i_y j_y} \sum_{m_y n_y k_y} \beta^y_{m_y n_y k_y} b_{m_y n_y}$$

The resulting algorithm xy:

$$[c_{p_{xy} q_{xy}}] = \sum_{k_{xy}} \gamma^{xy}_{p_{xy} q_{xy} k_{xy}} \sum_{i_{xy} j_{xy} k_{xy}} \alpha^{xy}_{i_{xy} j_{xy} k_{xy}} a_{i_{xy} j_{xy}} \sum_{m_{xy} n_{xy} k_{xy}} \beta^{xy}_{m_{xy} n_{xy} k_{xy}} b_{m_{xy} n_{xy}}$$

How to derive the coefficient matrices?

The following is not obvious, and a proof is omitted here. But in doubt, the resulting algorithms can be validated against Brent's equations.

Coefficient matrix γ_{xy}

$$\begin{aligned}
p_x &= 1 \cdots M_x \\
q_x &= 1 \cdots P_x \\
p_y &= 1 \cdots M_y \\
q_y &= 1 \cdots P_y \\
k_x &= 1 \cdots K_x \\
k_y &= 1 \cdots K_y \\
p_{xy} &= p_x M_y + p_y \\
q_{xy} &= q_x P_y + q_y \\
k_{xy} &= k_x K_y + k_y \\
\gamma^{xy}_{p_{xy} q_{xy} k_{xy}} &= \gamma^{x}_{p_x q_x} \gamma^{y}_{p_y q_y}
\end{aligned}$$

Coefficient matrix α_{xy}

$$\begin{aligned}
i_x &= 1 \cdots M_x \\
j_x &= 1 \cdots N_x \\
i_y &= 1 \cdots M_y \\
j_y &= 1 \cdots N_y \\
k_x &= 1 \cdots K_x \\
k_y &= 1 \cdots K_y \\
i_{xy} &= i_x M_y + i_y \\
j_{xy} &= j_x N_y + j_y \\
k_{xy} &= k_x K_y + k_y \\
\alpha^{xy}_{i_{xy} j_{xy} k_{xy}} &= \alpha^{x}_{i_x j_x k_x} \alpha^{y}_{i_y j_y k_y}
\end{aligned}$$

Coefficient matrix β_{xy}

$$\begin{aligned}
m_x &= 1 \cdots N_x \\
n_x &= 1 \cdots P_x \\
m_y &= 1 \cdots N_y \\
n_y &= 1 \cdots P_y \\
k_x &= 1 \cdots K_x \\
k_y &= 1 \cdots K_y \\
m_{xy} &= m_x N_y + m_y \\
n_{xy} &= n_x P_y + n_y \\
k_{xy} &= k_x K_y + k_y \\
\beta^{xy}_{m_{xy} n_{xy} k_{xy}} &= \beta^{x}_{m_x n_x k_x} \beta^{y}_{m_y n_y k_y}
\end{aligned}$$