Multiplication des matrices: Algorithme de Strassen

Nicolas Iung, Aurélien Blais

January 20, 2019

Chapter 1

Algorithme de Strassen

Question 1 8 multiplications :

$$c_{11} = a_{11} * b_{11} + a_{12} * b_{21} \tag{1.1}$$

$$c_{12} = a_{11} * b_{11} + a_{12} * b_{22} \tag{1.2}$$

$$c_{21} = a_{21} * b_{11} + a_{22} * b_{21} \tag{1.3}$$

$$c_{22} = a_{21} * b_{12} + a_{22} * b_{22} \tag{1.4}$$

Question 2 Développons les formules de Strassen :

$$c_{11} = q_1 - q_3 - q_5 + q_7$$
$$= [a_{11}b_{22} - a_{12}b_{22}] - [a_{22}b_{11} + a_{22}b_{21}] - [-a_{11}b_{11} + a_{11}b_{22} - a_{22}b_{11} + a_{22}b_{22}] + [a_{12}b_{21} - a_{12}b_{22} + a_{22}b_{21} - a_{22}b_{22}]$$
$$= -[-a_{11}b_{11} + a_{22}b_{22}] + [a_{12}b_{21} - a_{22}b_{22}]$$
$$= a_{11}b_{11} + a_{12}b_{21}$$

$$\begin{aligned} c_{12} &= a_{11}(b_{12} + b_{22}) - (a_{11} - a_{12})b_{22} \\ &= a_{11}b_{12} + a_{11}b_{22} - a_{11}b_{22} + a_{12}b_{22} \\ &= a_{11}b_{12} + a_{12}b_{22} \end{aligned}$$

$$\begin{aligned} c_{21} &= (a_{21} - a_{22})b_{11} + a_{22}(b_{11} + b_{21}) \\ &= a_{21}b_{11} - a_{22}b_{11} + a_{22}b_{11} + a_{22}b_{21} \\ &= a_{21}b_{11} + a_{22}b_{21} \end{aligned}$$

$$c_{22} = -(a_{21} - a_{22})b_{11} - a_{11}(b_{12} + b_{22}) + (a_{11} + a_{22})(b_{22} - b_{11}) + (a_{11} + a_{21})(b_{11} + b_{12})$$
$$= a_{21}b_{11} + a_{22}b_{11} - a_{11}b_{12} - a_{11}b_{12} + a_{11}b_{22} - a_{11}b_{11} + a_{22}b_{22} - a22b_{11} + a_{11}b_{11} + a_{11}b_{12} + a_{21}b_{11} + b_{21}b_{12}$$
$$= a_{21}b_{12} + a_{22}b_{22}$$

Les calculs effectués en partant des équations résultantes des formules de Strassen donnent bien le résultat de l'algorithme naïf de multiplication matricielle.

Question 3 Au vue de la question 2, on peut conclure que nous ne faisons qu'une multiplication par coefficient q donc un total de 7 multiplication. Cela nous fais donc économiser une multiplication.

Chapter 2

Produit matriciel par blocs

Question 1 En appliquant le calcul par blocs :

$$c_{11} = 1 * 1 + 2 * 2 + 0 * 0 + 1 * 1 = 6 \tag{2.1}$$

$$c_{12} = 1 * (-1) + 2 * 0 + 0 * 1 + 1 * 0 = -1 \tag{2.2}$$

$$c_{13} = 1 * 0 + 2 * 1 + 0 * 1 + 1 * 0 = 2 \tag{2.3}$$

$$c_{14} = 1 * 1 + 2 * 1 + 0 * 0 + 1 * (-1) = 2 \tag{2.4}$$

$$c_{21} = 3 * 1 + 4 * 2 + (-1) * 0 + 1 * 1 = 12 \tag{2.5}$$

$$c_{22} = 3 * (-1) + 4 * 0 + (-1) * 1 + 1 * 0 = -4 \tag{2.6}$$

$$c_{23} = 3 * 0 + 4 * 1 + (-1) * 1 + 1 * 0 = 3 \tag{2.7}$$

$$c_{24} = 3 * 1 + 4 * 1 + (-1) * 0 + 1 * (-1) = 6 \tag{2.8}$$

$$c_{31} = 1 * 1 + 0 * 2 + 1 * 0 + 2 * 1 = 3 \tag{2.9}$$

$$c_{32} = 1 * (-1) + 0 * 0 + 1 * 1 + 2 * 0 = 0 \tag{2.10}$$

$$c_{33} = 1 * 0 + 0 * 1 + 1 * 1 + 2 * 0 = 1 \tag{2.11}$$

$$c_{34} = 1 * 1 + 0 * 1 + 1 * 0 + 2 * (-1) = -1 \tag{2.12}$$

$$c_{41} = 0 * 1 + 1 * 2 + 3 * 0 + 4 * 1 = 6 \tag{2.13}$$

$$c_{42} = 0 * (-1) + 1 * 0 + 3 * 1 + 4 * 0 = 3 \tag{2.14}$$

$$c_{43} = 0 * 0 + 1 * 1 + 3 * 1 + 4 * 0 = 4 \tag{2.15}$$

$$c_{43} = 0 * 0 + 1 * 1 + 3 * 1 + 4 * 0 = 4 \tag{2.16}$$

Donc $C = \begin{array}{cc} & \begin{array}{cccc} c_1 & c_2 & c_3 & c_4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 6 & -1 & 2 & 2 \\ 12 & -4 & 3 & 6 \\ 3 & 0 & 1 & -1 \\ 6 & 3 & 4 & -3 \end{pmatrix} \end{array}$

Chapter 3

Algorithme de Strassen

Question 1 Formules de Strassen par blocs

$$q_1 = (A_{11} - A_{12}) * B_{22} \tag{3.1}$$

$$q_2 = (A_{21} - A_{22}) * B_{11} \tag{3.2}$$

$$q_3 = A_{22} * (B_{11} + B_{21}) \tag{3.3}$$

$$q_4 = A_{11} * (B_{12} + B_{22}) \tag{3.4}$$

$$q_5 = (A_{11} + A_{22}) * (B_{22} - B_{11}) \tag{3.5}$$

$$q_6 = (A_{11} + A_{21}) * (B_{11} + B_{12}) \tag{3.6}$$

$$q_7 = (A_{11} + A_{22}) * (B_{21} + B_{22}) \tag{3.7}$$

Question 2 Nombre de multiplications pour une matrice 2^k * 2^k

Comme vu précédemment, dans le cas de matrices de taille 2^1, on effectue 7 multiplications, correspondant aux calculs des coefficients q.

L'algorithme étant récursif, chaque étape nécessite le calcul des coefficients q et effectue donc 7 multiplications.

On peut donc en déduire que dans le cas de matrices de taille 2^2, nous effectuerons 7 * 7 multiplications, correspondant au 7 itérations des coefficients q pour les matrices de taille 2^1 et 7 itérations pour les matrices de taille 2^2.

On peut donc en déduire un cas général, de forme $u_k = 7u_{k-1}$

Question 3 Valeur de u_1

Pour u_1, nous ne calculons qu'une fois les coefficients q, soit 7 multiplications, $u_1 = 7$

Question 4 Déduire $u_k = 7^k$

Comme vu précédemment à la question 2, chaque itération demande d'effectuer 7 multiplications et dans le cas de u_1, nous avons aussi besoin de 7 multiplications.

$$\prod_{i=1}^{k} 7 = 7^k$$

Question 5 Montrer que $u_k = n^{\frac{ln(7)}{ln(2)}} \simeq n^{2,81}$

$$\begin{aligned} u_k &= 7^k \\ \iff 7^k &= n^\alpha \end{aligned}$$

$$\begin{aligned} ln(7) &= \alpha ln(n) \\ \iff ln(7) &= \alpha ln(2) \\ \iff \alpha &= \frac{ln(7)}{ln(2)} \end{aligned}$$

$$u_k = n^{\frac{ln(7)}{ln(2)}}$$

Question 6 Nombre de multiplications pour la méthode classique

Pour la multiplication classique de deux matrices A x B de tailles n x n, pour une colonne i et une ligne j, chaque élément u_{ij} est la somme du produit de chaque éléments de la ligne et de la colonne.

Exemple : $A : \begin{pmatrix} a & b \\ c & d \end{pmatrix} B : \begin{pmatrix} e & f \\ g & h \end{pmatrix} Resultat : \begin{pmatrix} a*e+b*g & a*f+b*h \\ c*e+d*g & c*f+d*h \end{pmatrix}$

On remarque alors que le nombre de multiplication pour chaque éléments correspond à la taille de la matrice. Il est ensuite répété pour chaque lignes et pour chaque colonnes.

$$n * n * n = \boldsymbol{n^3}$$

Question 7 Comparatif entre les algorithmes

Pour l'algorithme classique :

$$n^3 = 128^3 = 2097152$$

Pour l'algorithme de Strassen :

$$\begin{aligned} 128 &= 2^7 \\ \iff u_7 &= 7^7 = 823543 \end{aligned}$$

L'économie réalisée est donc de $2097152 - 823543 = 1273609$ multiplications

Chapter 4

Implémentation

Le benchmarking de l'algorithme à été effectué en executant les deux algorithmes sur des tableaux de tailles 2^k. Le résultat est le suivant.

k	1	2	4	8	16	32	64
Classique	$0,000042$	$0,000039$	$0,000118$	$0,000729$	$0,005180$	$0,037036$	$0,269134$
Strassen	$0,000010$	$0,000161$	$0,001028$	$0,008187$	$0,051670$	$0,361173$	$2,587751$

k	128	256	512	1024	2048
Classique	$2,302152$	$18,434917$	$148,468258$	$1143,45661$	$9189,38358$
Strassen	$18,336071$	$129,776649$	$878,765119$	$6224,82734$	$42726,193543$

Selon nos données, Strassen ne semble jamais surpasser l'algorithme classique. Au vu de ce résultat, nous pouvons déterminer que l'implémentation de notre algorithme souffre d'un problème d'optimisation, notamment dans l'initialisation des matrices de taille 2^{k-1}

L'algorithme de Strassen n'est utile que sur des matrices d'une taille suffisamment important. Pour des tailles plus petite, il est alors plus intéressant d'utiliser l'algorithme de multiplication classique.

Selon Wikipedia, il existe des algorithmes théoriquement plus performant que celui de Strassen, avec par exemple l'algorithme de Coppersmith-Winograd ayant une complexité en $O(n^{2,376})$. Cet algorithme n'est cependant pas utilisé, du fait qu'il ne serait optimal que sur des matrices ayant une taille gigantesque.