FFT 优化

刘润

2020/05/14

1 FFT 的优化

在傅里叶变换一文中，得知 DFT 可以由以下公式表示：

$$X(K) = W^{nk}x(n) \tag{1}$$

将矩阵展开之后即为：

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & \dots & W^0 \\ W^0 & W^{1\times1} & W^{2\times1} & \dots & W^{(N-1)\times1} \\ \vdots & \vdots & \vdots & & \vdots \\ W^0 & W^{1\times(N-1)} & W^{2\times(N-1)} & \dots & W^{(N-1)\times(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} \tag{2}$$

并且我们得知旋转因子 W_N^{nk} 有以下性质：

$$W^{nk} = W^{nk+N} \tag{3}$$

$$W^{nk+\frac{N}{2}} = -W^{nk} \tag{4}$$

根据式 (3) 和式 (4) 可以对 DFT 进行优化，下面以 8 点 DFT 为例进行说明。当 $N = 8$ 时，其旋转因子的矩阵为：

$$
\begin{array}{c}
 \\ n=0 \\ n=1 \\ n=2 \\ n=3 \\ n=4 \\ n=5 \\ n=6 \\ n=7
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} k=0 & k=1 & k=2 & k=3 & k=4 & k=5 & k=6 & k=7 \end{array} \\
\begin{bmatrix}
W^{0\times0} & W^{0\times0} & W^{0\times0} & W^{0\times0} & W^{0\times0} & W^{0\times0} & W^{0\times0} & W^{0\times0} \\
W^{1\times0} & W^{1\times1} & W^{1\times2} & W^{1\times3} & W^{1\times4} & W^{1\times5} & W^{1\times6} & W^{1\times7} \\
W^{2\times0} & W^{2\times1} & W^{2\times2} & W^{2\times3} & W^{2\times4} & W^{2\times5} & W^{2\times6} & W^{2\times7} \\
W^{3\times0} & W^{3\times1} & W^{3\times2} & W^{3\times3} & W^{3\times4} & W^{3\times5} & W^{3\times6} & W^{3\times7} \\
W^{4\times0} & W^{4\times1} & W^{4\times2} & W^{4\times3} & W^{4\times4} & W^{4\times5} & W^{4\times6} & W^{4\times7} \\
W^{5\times0} & W^{5\times1} & W^{5\times2} & W^{5\times3} & W^{5\times4} & W^{5\times5} & W^{5\times6} & W^{5\times7} \\
W^{6\times0} & W^{6\times1} & W^{6\times2} & W^{6\times3} & W^{6\times4} & W^{6\times5} & W^{6\times6} & W^{6\times7} \\
W^{7\times0} & W^{7\times1} & W^{7\times2} & W^{7\times3} & W^{7\times4} & W^{7\times5} & W^{7\times6} & W^{7\times7}
\end{bmatrix}
\end{array}
\tag{5}
$$

通过式 (3) 和式 (4) 可对式 (5) 的矩阵进行简化，例如将 W^5 化简为 $-W^1$，将 W^8 化简为 W^0。通过这两个公式，可以将以上矩阵化简为以下形式：

$$
\begin{array}{c}
 \\ n=0 \\ n=1 \\ n=2 \\ n=3 \\ n=4 \\ n=5 \\ n=6 \\ n=7
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} k=0 & k=1 & k=2 & k=3 & k=4 & k=5 & k=6 & k=7 \end{array} \\
\begin{bmatrix}
W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\
W^0 & W^1 & W^2 & W^3 & -W^0 & -W^1 & -W^2 & -W^3 \\
W^0 & W^2 & -W^0 & -W^2 & W^0 & W^2 & -W^0 & -W^2 \\
W^0 & W^3 & -W^2 & W^1 & -W^0 & -W^3 & W^2 & -W^1 \\
W^0 & -W^0 & W^0 & -W^0 & W^0 & -W^0 & W^0 & -W^0 \\
W^0 & -W^1 & W^2 & -W^3 & -W^0 & W^1 & -W^2 & W^3 \\
W^0 & -W^2 & -W^0 & W^2 & W^0 & -W^2 & -W^0 & W^2 \\
W^0 & -W^3 & -W^2 & -W^1 & -W^0 & W^3 & W^2 & W^1
\end{bmatrix}
\end{array}
\tag{6}
$$

从式 (6) 我们可以发现一系列规律。例如从行看来，奇数行的前半部分和后半部分是一致的；偶数行的前半部分和后半部分对应位置元素取负数即可。从列看来，以上矩阵也具有相同性质。从运算的本质来看，DFT 变为通过奇偶划分将矩阵运算变换为 FFT 多级蝶形运算，其本质就是利用旋转因子的周期性和对称性减少旋转因子的个数，将具有相同数值的旋转因子合并为一个旋转因子，进而减少乘法和加法运算。而以上的矩阵的优化就是对 FFT 的直观体现，在矩阵内部根据两个性质直接进行旋转因子的合并。其实式 (6) 中的矩阵和输入的时间序列相乘得到的就是 FFT 蝶形运算的输出结果。例如：

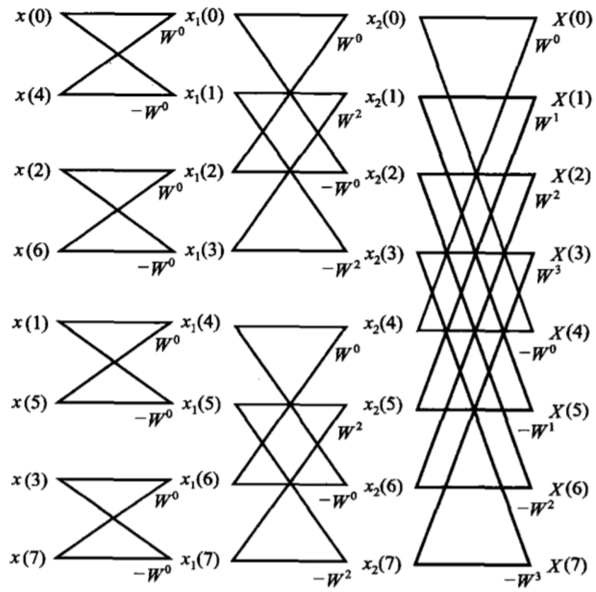

图 1: N=8 的 FFT 流程图

上图中 $X(5) = [x(0) - x(4)]W^0 + [-x(1) + x(5)]W^1 + [x(2) - x(6)]W^2 + [-x(3) + x(7)]W^3$，当使用式 (6) 中矩阵和输入序列相乘后得到的结果相同。所以我们可以得到各个输出点的情况：

$$X(0) = [x(0) + x(1) + x(2) + x(3) + x(4) + x(5) + x(6) + x(7)]W^0 \tag{7}$$

$$X(1) = [x(0)-x(4)]W^0+[x(1)-x(5)]W^1+[x(2)-x(6)]W^2+[x(3)-x(7)]W^3 \tag{8}$$

$$X(2) = [x(0) - x(2) + x(4) - x(6)]W^0 + [x(1) - x(3) + x(5) - x(7)]W^2 \tag{9}$$

$$X(3) = [x(0)-x(4)W^0+[x(3)-x(7)]W^1-[x(2)-x(6)]W^2+[x(1)-x(5)]W^3 \tag{10}$$

$$X(4) = [x(0) - x(1) + x(2) - x(3) + x(4) - x(5) + x(6) - x(7)]W^0 \tag{11}$$

$$X(5) = [x(0)-x(4)]W^0-[x(1)-x(5)]W^1+[x(2)-x(6)]W^2-[x(3)-x(7)]W^3 \tag{12}$$

$$X(6) = [x(0) - x(2) + x(4) - x(6)]W^0 - [x(1) - x(3) + x(5) - x(7)]W^2 \tag{13}$$

$$X(7) = [x(0)-x(4)]W^0-[x(3)-x(7)]W^1-[x(2)-x(6)]W^2-[x(1)-x(5)]W^3 \tag{14}$$

在式 (7) 到式 (14) 出现的输入序列组合分别为:

$S_1 = x(0) + x(1) + x(2) + x(3) + x(4) + x(5) + x(6) + x(7)$
$S_2 = x(0) - x(4)$
$S_3 = x(1) - x(5)$
$S_4 = x(2) - x(6)$
$S_5 = x(3) - x(7)$
$S_6 = x(0) - x(2) + x(4) - x(6)$
$S_7 = x(1) - x(3) + x(5) - x(7)$
$S_8 = x(0) - x(1) + x(2) - x(3) + x(4) - x(5) + x(6) - x(7)$
$S_8 = x(0) - x(1) + x(2) - x(3) + x(4) - x(5) + x(6) - x(7)$

转换为矩阵表达为:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} S_1 & 0 & 0 & 0 \\ S_2 & S_3 & S_4 & S_5 \\ S_6 & 0 & S_7 & 0 \\ S_6 & S_5 & -S_4 & S_2 \\ S_8 & 0 & 0 & 0 \\ S_2 & -S_3 & S_4 & -S_5 \\ S_6 & 0 & -S_7 & 0 \\ S_6 & -S_5 & -S_4 & -S_2 \end{bmatrix} \begin{bmatrix} W^0 \\ W^1 \\ W^2 \\ W^3 \end{bmatrix} \tag{15}$$

所以以上共实现了 10 种复数乘法，分别是:
S_1W^0
S_2W^0
S_3W^1
S_4W^2
S_5W^3
S_6W^0
S_7W^3
S_5W^1
S_2W^3
S_8W^1

共实现了 38 次加法，分别是:
式 (7):7 次
式 (8):7 次
式 (9):7 次
式 (10):3 次
式 (11):7 次
式 (12):3 次
式 (13):1 次
式 (14):3 次

2 编程实现

写了一个程序，可以实现旋转因子原始矩阵到优化矩阵的变换。程序路径为 77 服务器 $/home/liurun/lr/kws/test/fft/optimization.py$