Fast Fourier Transform

March 26, 2011

1 Definitions

1.1 The basic FFT

This note will discuss the fast Fourier transform (FFT). It is extracted from Henrici's paper. I will switch to his notation. Let $w_n = \exp(\frac{2\pi i}{n})$. Also suppose that all sequences are n-periodic, so $x_j = x_{j+n}$, and are defined for all n. We will denote the discrete Fourier transform by $\mathcal{F}_n$.

$$y_m = (\mathcal{F}_n x)_m = \frac{1}{n}\sum_{k=0}^{n-1} w_n^{-mk} x_k.$$

Now suppose $n = pq$. Then we divide the numbers components of x into p sets of q-vectors

$$x^{(j)} = (x_j, x_{j+p}, \ldots, x_{j+p(q-1)}),$$

$j = 0, 1, \ldots p-1$. Assume we know $\mathcal{F}_q x^{(j)}$ for $j = 0, 1, \ldots p-1$. Then we rewrite the formula for y_m, using $n = pq$.

$$y_m = \frac{1}{p}\sum_{j=0}^{p-1}\frac{1}{q}\sum_{k=0}^{q-1} w_n^{-m(j+ph)} x_{j+ph}.$$

Next notice $w_n^{-mj-mph} = w_n^{-mj} w_q^{-mh}$, since $w_n^p = \exp(\frac{2p\pi i}{pq}) = \exp(\frac{2\pi i}{q}) = w_q$. So

$$y_m = \frac{1}{p}\sum_{j=0}^{p-1} w_n^{-mj}(\frac{1}{q}\sum_{h=0}^{q-1} w_q^{-mh} x_{j+ph}).$$

Let

$$y_m^{(j)} = \frac{1}{q}\sum_{h=0}^{q-1} w_q^{-mh} x_{j+ph}.$$

Then

$$y^{(j)} = \mathcal{F}_q x^{(j)},$$

and

$$y_m = \frac{1}{p}\sum_{j=0}^{p-1} w_n^{-mj} y_m^{(j)}.$$

If we know the p q-vectors $y^{(j)}, j = 0, 1, \ldots, p-1$, then the cost of computing each component y_m is $p-1$ ops (we don't count multiplication by 1). There are n components of y so the cost is $n(p-1)$. If we stop at this point the cost of computing each $y^{(j)}$ is $(q-1)^2$ so the total is $n(p-1) + p(q-1)^2$. We over estimate

the last term with $pq(q-1) = n(q-1)$ and the cost is $n(p-1+q-1)$. Suppose we have a factorization $n = n_1 n_2 \dots n_\ell$. We continue this argument to find the cost is

$$n \sum_{i=1}^{\ell} (n_i - 1).$$

If $n = 2^\ell$ each $n_i = 2$ and the cost is

$$n\ell = n \log_2(n).$$

We can reduce this even further, if $p = 2$ in the initial discussion. The computation in equation (1) can be rewritten. Let $m = k + \ell q,\ k = 0, 1, \dots, q-1,\ \ell = 0, 1, \dots, p-1$ in equation (1) and in case $p = 2$, $p - 1 = 1$ Then

$$w_n^{-(k+\ell q)j} = w_2^{-\ell j} w_n^{-kj} = (-1)^{-\ell j} w_n^{-kj},$$

so

$$y_m = y_{k+\ell q} = \frac{1}{2}(y_k^{(0)} + (-1)^{-\ell} w_n^{-k} y_k^{(1)}), k = 0, 1, \dots, q-1,\ \ell = 0, 1.$$

and the only products that must be computed are $w_n^{-k} y_k^{(1)}$, $k = 0, 1, \dots, q-1$. (There are only two vectors $y^{(0)}, y^{(1)}$.) There are $q-1$ of these and they only need to be computed once. Before the cost of this stage was $n(p-1) = n$ if $p = 2$. Now it is $q - 1$ which we overestimate with $q = \frac{n}{2}$, half as many ops. This continues, to result in a cost of

$$\frac{n}{2} \log_2 n.$$

Here's another, recursive, way to describe the fft in the case $n = 2^\ell$. First let's assume that $n = 2m$. Then let $y = \mathcal{F}_n x$ so

$$\begin{aligned} n y_k =& x_0 + w_n^{-2k} x_2 + \dots w_n^{-(2m-2)k} x_{2m-2} && (1)\\ &+ w_n^{-k} x_1 + w_n^{-3k} x_3 + \dots w_n^{-(2m-1)k} x_{2m-1}. && (2) \end{aligned}$$

Suppose k is even, $k = 2q$. Then $w_n^{-jk} = w_n^{-j2q} = w_m^{-jq}$. To simplify the notation, let's replace w_n with w and let $\mu = w^2$ and $\mu^m = 1$. Now we can write

$$\begin{aligned} n y_k =& (x_0 + x_m w^{-mk}) + (x_1 w^{-k} + x_{m+1} w^{-(m+1)k}) + \dots && (3)\\ =& (x_0 + x_m \mu^{-mq}) + (x_1 \mu^{-q} + x_{m+1} \mu^{-(m+1)q}) + \dots && (4)\\ =& (x_0 + x_m) + (x_1 + x_{m+1}) \mu^{-q} + (x_2 + x_{m+2}) \mu^{-2q} + \dots && (5)\\ & && (6) \end{aligned}$$

Or also

$$y_{2q} = \frac{1}{m}[\frac{(x_0 + x_m)}{2} + \frac{(x_1 + x_{m+1})}{2} \mu^{-q} + \frac{(x_2 + x_{m+2})}{2} \mu^{-2q} + \dots]$$

Similarly

$$y_{2q+1} = \frac{1}{m}[\frac{(x_0 - x_m)}{2} + w^{-1} \frac{(x_1 - x_{m+1})}{2} \mu^{-q} + w^{-2} \frac{(x_2 - x_{m+2})}{2} \mu^{-2q} + \dots]$$

Let's denote x^+ by $x_j^+ = x_j + x_{j+m}$ and x^- by $x_j^- = w^{-j}(x_j - x_{j+m})$. Let's also write $y^e = [y_0, y_2, \dots y_{n-2}]$ and $y^o = [y_1, y_3, \dots, y_{n-1}]$. Then all of this can be written

$$y^e = \frac{1}{2} \mathcal{F}_m(x^+),\ 2y^o = \frac{1}{2} \mathcal{F}_m(x^-).$$

If we don't count the divisions by 2, the cost is just the cost of two computations of $\mathcal{F}_m$ and the cost of computing y^- which involves multiplying by m powers of w. Let $M(n)$ be the cost of computing $\mathcal{F}_n$. Then we have proved $M(2m) = 2M(m) + m$. Let's rewrite this as

$$\begin{aligned} M(n) =& 2M(\frac{n}{2}) + \frac{n}{2} && (7)\\ =& 2(2M(\frac{n}{2^2}) + \frac{n}{2^2}) + \frac{n}{2} && (8)\\ =& 2^2 M(\frac{n}{2^2}) + 2\frac{n}{2} && (9)\\ =& 2^3 M(\frac{n}{2^3}) + 3\frac{n}{2} && (10)\\ =& 2^{\ell-1} M(2) + (\ell-1)2^{\ell-1} && (11)\\ =& (1+\ell-1)2^{\ell-1} && (12)\\ =& \frac{n}{2}\log_2 n. && (13) \end{aligned}$$

1.2 Reversion Operator

We will find the *reversion operator* useful when we discuss convolutions.

Definition 1. *The* reversion operator R *is defined by*

$$(Rx)_m = x_{-m}$$

We have the following useful identities

$$\begin{aligned} (R\mathcal{F}_n x)_m =& \frac{1}{n}\sum_{k=0}^{n-1} w_n^{mk} x_k && (14)\\ (\mathcal{F}_n R x)_m =& \frac{1}{n}\sum_{k=0}^{n-1} w_n^{-mk} x_{-k} && (15)\\ =& \frac{1}{n}\sum_{k=0}^{n-1} w_n^{mk} x_k. && (16) \end{aligned}$$

Hence

$$\mathcal{F}_n R = R\mathcal{F}_n$$

and thus

$$n\mathcal{F}_n R = nR\mathcal{F}_n = \mathcal{F}^{-1}.$$

2 Applications

2.1 Convolutions

First we define a useful product that Henrici calls the *Hadamard product* and we denote it by a dot $\bullet$, $(x \bullet y)_k = x_k y_k$. Another multiplication, *convolution*, is denoted by $*$ is defined by $(x*y)_k = \sum_{j=0}^{n-1} x_j y_{k-j} = \sum_{j=0}^{n-1} y_j x_{k-j} = (y*x)_k$.

Theorem 1.

$$\mathcal{F}_n(x * y) = n\mathcal{F}_n x \bullet \mathcal{F}_n y \tag{17}$$
$$\mathcal{F}_n(x \bullet y) = \mathcal{F}_n x * \mathcal{F}_n y \tag{18}$$

Proof. Let

$$u = \mathcal{F}_n x,\ v = \mathcal{F}_n y.$$

Now

$$(\mathcal{F}_n(x \bullet y))_n = \frac{1}{n}\sum_{k=0}^{n-1} w^{-mk} x_k y_k.$$

By the inversion formula

$$y_k = (\mathcal{F}_n v)_k = \sum_{j=0}^{n-1} w^{kj} v_j.$$

So

$$(\mathcal{F}_n(x \bullet y))_m = \frac{1}{n}\sum_{k=0}^{n-1} w^{-mk} x_k (\sum_{j=0}^{n-1} w^{kj} v_j) \tag{19}$$
$$= \sum_{j=0}^{n-1} v_j (\frac{1}{n}\sum_{k=0}^{n-1} w^{-(m-j)k} x_k) \tag{20}$$
$$= \sum_{j=0}^{n-1} v_j u_{m-j} \tag{21}$$
$$= (\mathcal{F}_n y * \mathcal{F}_n x)_m \tag{22}$$
$$= (\mathcal{F}_n x * \mathcal{F}_n y)_m \tag{23}$$

The first statement of the theorem will be proved using the reversion operator. In this last formula, let $x = \mathcal{F}_n^{-1} u,\ y = \mathcal{F}_n^{-1} v$. Then we get

$$u * v = \mathcal{F}_n(\mathcal{F}_n^{-1} u \bullet \mathcal{F}_n^{-1} v) \tag{24}$$
$$= n^2 \mathcal{F}_n(R\mathcal{F}_n u \bullet R\mathcal{F}_n v) \tag{25}$$
$$= n^2 \mathcal{F}_n R(\mathcal{F}_n u \bullet \mathcal{F}_n v) \tag{26}$$
$$= n^2 R\mathcal{F}_n(\mathcal{F}_n u \bullet \mathcal{F}_n v) \tag{27}$$
$$= n\mathcal{F}_n u \bullet \mathcal{F}_n v, \tag{28}$$

since

$$n\mathcal{F}_n R = nR\mathcal{F}_n = \mathcal{F}^{-1}.$$

□

Corollary 1.

$$u * v = n^2 R\mathcal{F}_n(\mathcal{F}_n u \bullet \mathcal{F}_n v)$$

Corollary 2. *If $n = 2^\ell$ the convolution of two sequences can be computed by taking three discrete Fourier transforms via the fft and one Hadamard product. The cost is no more than*

$$\frac{3n}{2}\log_2(n) + n = \frac{3n}{2}\log_2(n) + n\log_2(2) < \frac{3n}{2}\log_2(2n)$$

complex multiplications.

2.2 Multiplying Polynomials and Large Integers