Subject: Re: Convolving speed issue
Posted by pgrigis on Thu, 17 Apr 2008 17:16:05 GMT
View Forum Message <> Reply to Message

I haven't followed exactly what you are doing below,
and I am not familiar with Matlab's conv2, but
couldn't you use the FFT method instead?
That should be much faster.

Paolo

rog...@gmail.com wrote:

> Hi there,
> I have a strange problem with the IDL convolving possibilities. I'd
> like to make a script which convolves 2 matrices - e.g. a and b - in
> the same behavior conv2(a,b,'same') Matlab does. The Problem is not to
> make such a script, the problem is that IDL takes too long or hangs
> when i try to convolve larger matrices. I tried certainly all kinds of
> using the built-in IDL-convol method, but convolving large arrays ends
> always in different results compared to the very fast Matlab conv2.
> Maybe someone could help me. Here's the sample code:

```
>
> function size_dim, in, direction
> dims = size(in, /dimensions)
> return, dims[direction]
> end
>
> function zeropadding, in,xsize,ysize
> ;only for 2D-arrays
> xsize_in=size_dim(in,0)
> ysize_in=size_dim(in,1)
> shiftx  =  ceil((xsize-xsize_in)/2)
> shifty  =  ceil((ysize-ysize_in)/2)
> temp    =  fltarr(xsize,ysize)
> temp[0:xsize_in-1,0:ysize_in-1]  = in
> temp  =  shift(temp,shiftx,shifty)
> return, temp
> end
>
> function conv2, a,b
> size_a=[size_dim(a,0),size_dim(a,1)]
> size_b=[size_dim(b,0),size_dim(b,1)]
> a=zeropadding(a,size_a[0]+size_b[0], size_a[1]+size_b[1])
> b=zeropadding(b,size_a[0]+size_b[0], size_a[1]+size_b[1])
> c=fltarr(size_a[0]+size_b[0], size_a[1]+size_b[1], /nozero)
> addx  =  floor(total(size_a)/4)
> endx  =  ceil(double(total(size_a)/4))
> addy  =  floor(total(size_b)/4)
>  endy  =  ceil(double(total(size_b)/4))
>
>  for n1=0,size_a[0]+size_b[0]-1 do begin
>    for n2=0,size_a[1]+size_b[1]-1 do begin
>      temp=0
>      temp2=0
>      for k1=0+addx,size_a[0]+size_b[0]-1-endx do begin
>        for k2=0+addx,size_a[1]+size_b[1]-1-endy do begin
>          if n1-k1 gt-1 and n2-k2 gt-1 then begin
>            temp=a[k1,k2]*b[n1-k1+addx,n2-k2+addy]
>            temp2=temp2+temp
>          endif
>        endfor
>      endfor
>      c[n1,n2]=temp2
>    endfor
>  endfor
>  temp  =  shift(c,-2*addx,-2*addy)
>  return, temp[0:size_a[0]-1,0:size_b[0]-1]
>  end
>
>
>  pro conv
>  ; sample matrix -> magic(5) in Matlab
>  a=    [[17,    24,     1,     8,    15],$
>        [23,     5,     7,    14,    16],$
>         [4,     6,    13,    20,    22],$
>        [10,    12,    19,    21,     3],$
>        [11,    18,    25,     2,     9]]
>  b=2*a
>  c=a
>  d=b
>  print, 'Trying own convolution...',string(10b),conv2(a,b), string(10B)
>  print, 'Trying built in convolution...',string(10b),$
>  shift(convol(zeropadding(c,10,10),zeropadding(d,10,10), center=0,/
>  edge_wrap),-4,-4),string(10b)
>  end
>
```

> Maybe there is a solution for using reform in some way? It seems to be
> quicker as for-loops. But I can't imagine how it could work when the
> indices of the multiplying matrices are varying.
>
> Hope on help
>
> Thank you and best regards
>
> Chris

Subject: Re: Convolving speed issue
Posted by rogass on Thu, 17 Apr 2008 18:54:16 GMT

View Forum Message <> Reply to Message

Yes, you're right. But I have to produce the same results Matlab produces with conv2. The Implementation of convolving matrices in the frequency domain is unfortunately also different. I tried this before - for small and also for large matrices.

I can't understand, why there should not be another possibility within IDL to compute simple c[n1,n2]=sum(sum(a[k1,k1]*b[n1-k1,n1-k1]))? Any other ideas?

Nevertheless, thanks Paolo for the hint.

Best regards

Chris

Subject: Re: Convolving speed issue
Posted by pgrigis on Thu, 17 Apr 2008 20:03:39 GMT

View Forum Message <> Reply to Message

Again, I don't know for sure what Matlab does,
but the following gives the convolution of a * b

```
s1=size(a)
s2=size(b)
nx1=s1[1]
ny1=s1[2]
nx2=s2[1]
ny2=s2[2]

aa=fltarr(nx1+nx2-1,ny1+ny2-1)
bb=fltarr(nx1+nx2-1,ny1+ny2-1)
aa[0,0]=a
bb[nx1-1,ny1-1]=b

conv=double(shift(fft(fft(aa,-1)*fft(bb,-1),
1)*n_elements(aa),nx2,ny2))
;use double because real_part seem to be broken on my system for some
reaon...
```

Ciao,
Paolo

rog...@googlemail.com wrote:

> Yes, you're right. But I have to produce the same results Matlab
> produces with conv2. The Implementation of convolving matrices in the
> frequency domain is unfortunately also different. I tried this before
> - for small and also for large matrices.
>
> I can't understand, why there should not be another possibility within
> IDL to compute simple c[n1,n2]=sum(sum(a[k1,k1]*b[n1-k1,n1-k1]))? Any
> other ideas?
>
> Nevertheless, thanks Paolo for the hint.
>
> Best regards
>
> Chris

Subject: Re: Convolving speed issue
Posted by pgrigis on Thu, 17 Apr 2008 20:42:31 GMT

View Forum Message <> Reply to Message

As an option you may try to reduce the number of for loops,
something like the following should do it:

```
a=[[1,2,3],[4,5,6]]
b=[[1,0,0],[1,1,1],[2,1,1]]

s1=size(a)
nx1=s1[1]
ny1=s1[2]
s2=size(b)
nx2=s2[1]
ny2=s2[2]

res=fltarr(nx1+nx2-1,ny1+ny2-1)

FOR j=0,ny2-1 DO BEGIN
  FOR i=0,nx2-1 DO BEGIN
    res[i:i+nx1-1,j:j+ny1-1]+=a*b[i,j]
  ENDFOR
ENDFOR
```

Ciao,
Paolo

rog...@gmail.com wrote:
> Hi there,

> I have a strange problem with the IDL convolving possibilities. I'd
> like to make a script which convolves 2 matrices - e.g. a and b - in
> the same behavior conv2(a,b,'same') Matlab does. The Problem is not to
> make such a script, the problem is that IDL takes too long or hangs
> when i try to convolve larger matrices. I tried certainly all kinds of
> using the built-in IDL-convol method, but convolving large arrays ends
> always in different results compared to the very fast Matlab conv2.
> Maybe someone could help me. Here's the sample code:
>
> ```
> function size_dim, in, direction
> dims = size(in, /dimensions)
> return, dims[direction]
> end
>
> function zeropadding, in,xsize,ysize
> ;only for 2D-arrays
> xsize_in=size_dim(in,0)
> ysize_in=size_dim(in,1)
> shiftx  =  ceil((xsize-xsize_in)/2)
> shifty  =  ceil((ysize-ysize_in)/2)
> temp    =  fltarr(xsize,ysize)
> temp[0:xsize_in-1,0:ysize_in-1]  = in
> temp  =  shift(temp,shiftx,shifty)
> return, temp
> end
>
> function conv2, a,b
> size_a=[size_dim(a,0),size_dim(a,1)]
> size_b=[size_dim(b,0),size_dim(b,1)]
> a=zeropadding(a,size_a[0]+size_b[0], size_a[1]+size_b[1])
> b=zeropadding(b,size_a[0]+size_b[0], size_a[1]+size_b[1])
> c=fltarr(size_a[0]+size_b[0], size_a[1]+size_b[1], /nozero)
> addx  =  floor(total(size_a)/4)
> endx  =  ceil(double(total(size_a)/4))
> addy  =  floor(total(size_b)/4)
> endy  =  ceil(double(total(size_b)/4))
>
> for n1=0,size_a[0]+size_b[0]-1 do begin
>   for n2=0,size_a[1]+size_b[1]-1 do begin
>     temp=0
>     temp2=0
>     for k1=0+addx,size_a[0]+size_b[0]-1-endx do begin
>       for k2=0+addx,size_a[1]+size_b[1]-1-endy do begin
>         if n1-k1 gt-1 and n2-k2 gt-1 then begin
>           temp=a[k1,k2]*b[n1-k1+addx,n2-k2+addy]
>           temp2=temp2+temp
>         endif
>       endfor
> ```

```
>     endfor
>     c[n1,n2]=temp2
>   endfor
> endfor
> temp  =  shift(c,-2*addx,-2*addy)
> return, temp[0:size_a[0]-1,0:size_b[0]-1]
> end
>
>
> pro conv
> ; sample matrix -> magic(5) in Matlab
> a=    [[17,    24,     1,     8,    15],$
>        [23,     5,     7,    14,    16],$
>         [4,     6,    13,    20,    22],$
>        [10,    12,    19,    21,     3],$
>        [11,    18,    25,     2,     9]]
> b=2*a
> c=a
> d=b
> print, 'Trying own convolution...',string(10b),conv2(a,b), string(10B)
> print, 'Trying built in convolution...',string(10b),$
> shift(convol(zeropadding(c,10,10),zeropadding(d,10,10), center=0,/
> edge_wrap),-4,-4),string(10b)
> end
>
```

> Maybe there is a solution for using reform in some way? It seems to be
> quicker as for-loops. But I can't imagine how it could work when the
> indices of the multiplying matrices are varying.
>
> Hope on help
>
> Thank you and best regards
>
> Chris

Subject: Re: Convolving speed issue
Posted by rogass on Fri, 18 Apr 2008 08:13:47 GMT

View Forum Message <> Reply to Message

Thank you again, Paolo. Your code works better and much faster than
mine one- but unfortunately only for small matrices. I tried to build
the code for the method presented here:

 http://www.mathworks.com/access/helpdesk/help/techdoc/index.
html?/access/helpdesk/help/techdoc/ref/conv2.html&http:/ /www.google.com/search?q=conv2

I don't understand, why it works for small matrices and not for big

ones. Maybe its depending on how differ the number of elements in each
dimension of each matrix.

The FFT method has also the same problem.

Any other solutions or recommendations, why it won't work with medium
matrices e.g. a[0:35,0:59] and b[0:59,0:119]

Maybe with reform(rebin...

Thanks and best regards

Chris

Subject: Re: Convolving speed issue
Posted by rogass on Fri, 18 Apr 2008 09:16:53 GMT
View Forum Message <> Reply to Message

Thank you again, Paolo. Your code works better and much faster than
mine one- but unfortunately only for small matrices. I tried to build
the code for the method presented here:

http://www.mathworks.com/access/helpdesk/help/techdoc/index.
html?/access/helpdesk/help/techdoc/ref/conv2.html&http:/ /www.google.com/search?q=conv2

I don't understand, why it works for small matrices and not for big
ones. Maybe its depending on how differ the number of elements in each
dimension of each matrix.

The FFT method has also the same problem.

Any other solutions or recommendations, why it won't work with medium
matrices e.g. a[0:35,0:59] and b[0:59,0:119]

Maybe with reform(rebin...

Thanks and best regards

Chris

Subject: Re: Convolving speed issue
Posted by pgrigis on Fri, 18 Apr 2008 13:17:25 GMT
View Forum Message <> Reply to Message

rog...@googlemail.com wrote:

> Thank you again, Paolo. Your code works better and much faster than
> mine one- but unfortunately only for small matrices.

Can you specify what problems you are having with large matrices?
Do you get an IDL error, or it is just too slow for you or what?

Ciao,
Paolo

> I tried to build
> the code for the method presented here:
>
> http://www.mathworks.com/access/helpdesk/help/techdoc/index.
html?/access/helpdesk/help/techdoc/ref/conv2.html&http:/ /www.google.com/search?q=conv2
>
> I don't understand, why it works for small matrices and not for big
> ones. Maybe its depending on how differ the number of elements in each
> dimension of each matrix.
>
> The FFT method has also the same problem.
>
> Any other solutions or recommendations, why it won't work with medium
> matrices e.g. a[0:35,0:59] and b[0:59,0:119]
>
> Maybe with reform(rebin...
>
> Thanks and best regards
>
> Chris

Subject: Re: Convolving speed issue
Posted by rogass on Mon, 21 Apr 2008 15:08:57 GMT

View Forum Message <> Reply to Message

Dear Paolo,
it's unfortunately just too slow, so I'm just trying to enhance the
speed of your well working method. Maybe you have further ideas?

Thanks and best regards

Christian