Subject: Re: array operations
Posted by Spon on Wed, 12 Sep 2007 09:32:01 GMT
View Forum Message <> Reply to Message

On Sep 12, 9:13 am, pa...@gmx.de wrote:

> Hello,
> I have a small question to an array operation.
>
> I have a 3-dimensional array. 10x10x100 ... the first 2 dimensions 10x10 are spatial dimensions (so one image with 10 width with and 10 pix height). The third dimension is the time dimension. So every images was acquired 100 times.
>
> What I wanted to do is now to compute the the mean of every pixel in time dimension.
>
> So for one special pixel (e.g. [3,3]) i would write
>
> meanpixel = mean(myarray[3,3,*])
>
> but how is it, if i would like to do this operation for every pixel in the spatial dimensions? I just saw a possibility with a for loop.
>
> like
>
> ```
>       FOR j = 0, 9 DO BEGIN
>
>             FOR k = 0, 9 DO BEGIN
>
>                   meanarray[i,j]=mean(myarray[i,j,*])
>
>             ENDFOR
>
>       ENDFOR
> ```
>
> This isn't a very fast possibility especially when the array goes big, and i have to do that operation for a full image frame which is 1024x1024 pixel.
>
> Is there maybe another way how to act with it? I know that IDL is very strong with array operations, so maybe there is any another solution, which maybe doesn't need the two loops. Thanks a lot for your kind responses.
>
> greetings
>
> martin

Hello Martin, hello list!

Apologies in advance for any glaring errors, this is my first post...

You may wish to make use of the fact that, while the 'MEAN' function
has no [DIMENSIONS] keyword, the 'TOTAL' function does. Why? I'm not
really sure.

e.g.:

```
s=size(myarray)

meandata=(TOTAL(myarray,3,/NAN) / s(3))
```

I'm not sure this is any faster, but I suspect so. Another thing you
might consider if you've got large datasets of floating points, is to
bytescale (BYTSCL) your data first; I guess it depends how important
your raw absolute data are to you, or if you're looking at relative
rather than absolute values and can afford to shrink down your dataset
before you begin juggling with it.

Also, not directly related, but worth a careful read or 16: J. D.
Smith's amazing dimension juggling tutorials:
http://www.dfanning.com/tips/rebin_magic.html
http://www.dfanning.com/tips/array_concatenation.html

plus, of course, anything and everything written by the brilliant Dave
Fanning without whom I'd still be trapped in a FOR loop myself :-)

Hope this helps,

Chris

PS I tried playing around with the PRESERVE_TYPE keyword, but just
like the helpfile predicted I got wild overflows and complete
gibberish in the output (my raw data is currently single-precision
float).

Subject: Re: array operations
Posted by Craig Markwardt on Wed, 12 Sep 2007 10:18:43 GMT

View Forum Message <> Reply to Message

payon@gmx.de writes:
> Hello,
> I have a small question to an array operation.
>
> I have a 3-dimensional array. 10x10x100 ... the first 2 dimensions
> 10x10 are spatial dimensions (so one image with 10 width with and 10

> pix height). The third dimension is the time dimension. So every
> images was acquired 100 times.
>
> What I wanted to do is now to compute the the mean of every pixel in
> time dimension.
>
> So for one special pixel (e.g. [3,3]) i would write
>
> meanpixel = mean(myarray[3,3,*])
>
> but how is it, if i would like to do this operation for every pixel in
> the spatial dimensions? I just saw a possibility with a for loop.
>

If you are just computing the mean value, then using TOTAL is the fastest. You can sum over any axis you wish, and then divide by the number of pixels in the sum.

For example,

```
  mean_image = total(myarray,3) / 100
```

would give the mean image. The "3" tells TOTAL to sum over the third dimension.

Good luck,
Craig

--
 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------

Subject: Re: array operations
Posted by Steve[5] on Wed, 12 Sep 2007 10:59:26 GMT

View Forum Message <> Reply to Message

Since this sounds something like some of the processing I do, have you considered a different "average" - the median, idl's median function does have a dimension parameter, so:-

```
medpixel=median(myarray,dimension=3,/even)
```

Steve

Subject: Re: array operations
Posted by Conor on Wed, 12 Sep 2007 15:11:20 GMT
View Forum Message <> Reply to Message

The other way to calculate the mean is by using rebin. When rebin shrinks an array, it takes a mean of values. So rather than using:

the general way of doing it is;

```
mean_image = rebin(myarray,ncols,nrows)
```

or in your case:

```
mean_image = rebin(myarray,10,10)
```

rebin shrinks the array down to new dimensions of size 10x10. It averages along the third dimension when making the new array.

On Sep 12, 6:59 am, Steve <f...@k.e> wrote:
> Since this sounds something like some of the processing I do, have you
> considered a different "average" - the median, idl's median function
> does have a dimension parameter, so:-
>
> medpixel=median(myarray,dimension=3,/even)
>
> Steve