Hi all! Welcome back to Kesho AI

Its January so let’s do something for Christmas.

We have a reinforcement learning project that we will post soon, but today let’s look at a coding problem.

I was watching a Christmas show with my daughter. And cookie monster (show pic) was doing the twelve days of Christmas but with cookies. As I was sitting there I was thinking wow that is a lot of cookies.

So the coding problem is calculate how many cookies cookie monster ate. Don’t worry it will be a bit more in depth than it sounds.

First a quick note on for loops.

So when I was working on a research project I had a student write some of the code. One day she came up to me and apologized because her code had a for loop in it.

I know that sometimes for loops can be a sign of a beginner programmer. It really is about context.

I do a lot of data science and since I am not sharing my code and I only have to run it a few times, if a for loop seems clearer to me I will use one.

However, if you are working in groups where performance and readability come into play then you want to remove loops when possible.

Okay back to the coding problem. We will approach this like an interview question and come up with one bad solution, one good solution and one great solution.

Another caveat, and this is for the people who watched that weird Eddie Murphy movie Candy cane lane, we will be counting every time they repeat. So instead of 1 +2+ 3 and so on

The problem becomes 1 + (1+2) + (1+2+3).

Ok so lets look at this mathematically. It is pretty straight forward as it is a sum of sums. (draw on white board) The bad solution to this will just use nested for loops (ouch painful to do) This is On2.

Write method a with time it.

Let’s see if we can get rid of one of these loops. So there is actually a pretty famous formula for the summing of counting numbers. The story goes that when mathematician Karl Gauss was a student hhis teacher asked him to sum all off the numbers up to 100. The teacher expected it to take the afternoon but Karl calculated it in his head and shouted out the solution almost immediately. This is the gauss summation. It means that sigma j =( n\*(n+1))/2. Lets add this in as method b and see if it speeds things up a bit. Great!

Ok well we have one more loop left. We could just write this as a single line using a list comprehension but let’s see if we can derive our own function. Lets first plot out our xs and ys up to 100 using matplot lib and see what it looks like. Ok, that almost looks like an exponential curve, but not quite. Let’s try to fit a polynomial to this using the numpy library. We will set this to 3 (because I trial and errored this and it worked the best). Great lets look at our coefficients. Ok so we have three pretty neat coefficients although 2 of the numbers are irrational. .16667 is essentially (1/6) .5 is ½ and .333 is 1/3. Now we can get rid of the second loop. I am going to use fraction here which will make the run slower but if you use the actual values it will run faster but be imprecise. Let’s call this method c and see how fast it goes. Yep the irrational numbers make it a bit slower but neater all around.

There you have it. Cookie monster ate x cookies. Hope you enjoyed this and we will see you next time.