#25 Fix the temporal pooler algorithm

Milestone: [1.0](http://sourceforge.net/p/openhtm/todo/milestone/1.0)

Status: accepted

Owner: nobody

Labels: None

Updated: 2013-05-31

Created: 2013-04-07

Creator: [Itay](http://sourceforge.net/u/itay2541/)

Private: No

As specified in this post : <https://sourceforge.net/p/openhtm/discussion/htm/thread/6de01744/?limit=25#5fe8/fcdd/908f/d013/300b/35a8>

Barry said he would investigate soon. Barry and I will then try to understand and to debug further what makes the temporal pooler to not work correctly.  
Will update also on the team forum on further details..

Discussion

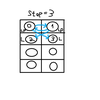
*   
  [Itay](http://sourceforge.net/u/itay2541/)   
  2013-04-08

I found a possible bug..  
When presenting the input "AAAX", at step = 3, a strange situation when one cell in one column having a synapse towards the other cell at the other column, and vice versa. this happens because both cells are selected as learning, and then selected as learners again at the next timestep. see included picture.  
I think this is a problematic situation, and why is that? because both cells (at the included picture 0 and 1) are also pointing towards other cells (2 and 3) which were made in a different timestep. eventually, when the columns will be active, cells 0 and 1 will feed each other in a recurrent way and will not allow the column to stop predicting itself - which might also hurt other connections that are good ones.  
I did not investigate to the bottom of it, but this looks like a potential bug, and one of the ways for a solution is to add the following rule : "select a learning cell from the cells from only those who were not learning cells at the last timestep"

will keep you up to date

**Attachments**

Top of Form

[](http://sourceforge.net/p/openhtm/todo/_discuss/thread/52260fdf/d3ef/attachment/possible_bug.png)  
[possible\_bug.png](http://sourceforge.net/p/openhtm/todo/_discuss/thread/52260fdf/d3ef/attachment/possible_bug.png) 

Bottom of Form

[Link](http://sourceforge.net/p/openhtm/todo/25/#d3ef)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* +   
    [Doug King](http://sourceforge.net/u/dwking/)   
    2013-04-08

I hope you are right. That would explain the current CLA behavior.

Can you tell us if you think this is a bug that we introduced or a bug in the CLA paper / pseudo-code? I, like you, have some doubts that the CLA paper code is correct, but I think the theory and the rest of the paper is valid.

[Link](http://sourceforge.net/p/openhtm/todo/25/#d3ef/64a4)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + -   
      [Itay](http://sourceforge.net/u/itay2541/)   
      2013-04-08

This doesn't appear at the paper..  
This is just the tip of the iceberg man, I haven't even did a full input circulation..  
I clearly think there are few things at the paper that are ignored or misused.  
One of the other things, is for an example, an update list for a neuron's segment, which is remembered for eternity until a cell is chosen as learning, and then executed.  
stuff such as segment update list, which reminds me more of a course in data structures rather than a biological neural network, a split state algorithm which uses massive matrix to manage its states, or even, calculating and passing up markov chains, as proposed by dileep george in his thesis - and even made up a full biological model for that.  
I don't think nature wants to build complex things, and if it must, then it will use the simplest thing / a slightly altered thing / a replica. so that if you look at my prototype algorithm at the team forum, you see how simple it is, yet suprisingly brought more results so far than the current temporal implementation which is broken at the moment.. but I hope that the temporal implementation will succseed and amaze me.

Last edit: Itay 2013-04-08

[Link](http://sourceforge.net/p/openhtm/todo/25/#d3ef/64a4/27dd)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + - *   
        [Nick](http://sourceforge.net/u/disturbed-ua/)   
        2013-04-08

an update list for a neuron's segment, which is remembered for eternity until a cell is chosen as learning, and then executed.

So it should be cleared on every time step?  
Did you notice that AAAX is actually AAAAX? But that doesn't change things.

Last edit: Nick 2013-04-08

[Link](http://sourceforge.net/p/openhtm/todo/25/#d3ef/64a4/27dd/ea30)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + - * +   
          [Itay](http://sourceforge.net/u/itay2541/)   
          2013-04-13

I have no idea about the real rules who can learn sequences correctly.. sorry  
and yes I have noticed, it has no real effect since it's a complex context forking either way

[Link](http://sourceforge.net/p/openhtm/todo/25/#d3ef/64a4/27dd/ea30/fed4)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

*   
  [Itay](http://sourceforge.net/u/itay2541/)   
  2013-04-13

Another thing I have discovered now :  
The most minimal way to represent the ABBCBB input, that contain a context forking, is like this :  
100 (A)  
010 (B)  
010 (B)  
001 (C)  
010 (B)  
010 (B)

The rules of the current implementation allow for a minimal representation of input using two active columns (this is because of the rule that says "do not form a segment to the same column"). so an input like "ABBCBB" (which was solved by Barry) can look like this :  
110000 (A)  
001100 (B)  
001100 (B)  
000011 (C)  
001100 (B)  
001100 (B)  
The current implementation was supposed to be able to learn this.  
However, to my suprise I have discovered that it can't learn and instead gets confused.  
The actual minimality for the current implementation to learn the "ABBCBB" input, which was solved by Barry, is 5 active columns to present an input. like this :  
111110000000000 (A)  
000001111100000 (B)  
000001111100000 (B)  
000000000011111 (C)  
000001111100000 (B)  
000001111100000 (B)  
My thoughts are unsure about this. First, it's very good that the temporal pooler managed to learn the "ABBCBB" input that contains a real context forking.  
However, I wonder why it can only learn when the minimal number of active columns for every input must be larger than five. The answers found can have a real impact. I must say that this minimal requirement can really make the debugging a hard business (I have no idea how to debug this in manageable complexity and time) and makes me wonder what other effects this limitation can have.

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* +   
    [Itay](http://sourceforge.net/u/itay2541/)   
    2013-04-13

Also, the number of new synapses must be >= the number of active synapses threshold + 1. otherwise it wouldn't work.  
Also, the number of new synapses can't be 1, even when the bug at line #416 at cell.cs is fixed and the active synapses threshold would truly be 1. it must always be bigger or equal than 2.

Last edit: Itay 2013-04-13

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + -   
      [Itay](http://sourceforge.net/u/itay2541/)   
      2013-04-13

Looks like the number of active columns per input letter must around bigger or equal than new synapses count + 1, and not limited to 5.

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + - *   
        [Itay](http://sourceforge.net/u/itay2541/)   
        2013-04-13

Alright guys.  
It looks like that with those parameters the performance is much better :  
segment active threshold = 1  
new number synapses = 3-5  
connected permanence = 0.2  
initial permanence = 0.25  
permanence increase = 0.01  
permanence decrease = 0.015  
spatial hardcoded = yes  
localality radius = 0  
you need to filter in the input vs predictions tab, by the most intense columns (to about 40% usually depending on the input) in order to see it.  
it's not anywhere perfect, but it's much better than previous tests I've done and comes close to the performance of my algorithm in simple tests.  
if your'e not using these settings you will not get good performance.

Last edit: Itay 2013-04-13

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + - * +   
          [Nick](http://sourceforge.net/u/disturbed-ua/)   
          2013-04-13

Thank you, I will take a look

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540/4855)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + - * +   
          [Nick](http://sourceforge.net/u/disturbed-ua/)   
          2013-04-13

What do you mean by saying "the performance is much better"?  
On this stage we can share projects with input files to understand full context of provided results.

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540/e5f8)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

  
[Itay](http://sourceforge.net/u/itay2541/)   
2013-04-13

I meant if you filter by 40% intense columns, and put any kind of sequence through it, then the performance you will get (input vs predictions when filtering 40% intense columns) using the settings I posted will be better than other kind of settings..

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540/e5f8/29d4)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

  
[Nick](http://sourceforge.net/u/disturbed-ua/)   
2013-04-13

any kind of sequence

OK, got it.

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540/e5f8/29d4/59bb)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

  
[Nick](http://sourceforge.net/u/disturbed-ua/)   
2013-04-13

It seems there is a problem in determining right or wrong predictions. Take a look at 3D picture: different sublayers learn the same input in correspondent contexts and can be as well right or wrong while being just black on 2D.  
I also don't get it why different columns of the same input get different permanences?

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540/e5f8/29d4/8199)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

  
[Itay](http://sourceforge.net/u/itay2541/)   
2013-04-13

I didn't understand you to the fullest  
I don't think different columns of the same input should get different permanences if the locality is 0.  
Your'e right in saying it predicts all the options. But if you want to see the true predicted option, then you must filter by columns intensity to some degree

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540/e5f8/29d4/8199/9605)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

  
[Nick](http://sourceforge.net/u/disturbed-ua/)   
2013-04-14

I say, if filter to lower percentage, some columns disappear because something is different for them (maybe not permanence but your 'strength' is).

if you want to see the true predicted option, then you must filter by columns intensity to some degree

you won't always notice if it's really true, it might be different cell from the same column

Last edit: Nick 2013-04-14

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540/e5f8/29d4/8199/9605/f047)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

*   
  [Itay](http://sourceforge.net/u/itay2541/)   
  2013-04-14

yes, some columns disappear because their prediction is not strong enough, depending on the precentage you set. also, there is a phenomena of a partial filtering of columns from the same permanences. this is because in the filtering loop, I go from i = 0 to i = index of filtering precentage. however, this might not be the best option, because some columns who have the same permanence appear or disappear halfway (they are not disappearing or appearing at once), and this can be worked on so that the filtering precentage will be show / not show all the columns of the same permanence. it's a pretty easy fix for the filtering operation.

"you won't always notice if it's really true, it might be different cell from the same column" - true, but it doesn't really matter since all you see, and all what the next region is seeing, is the output of the columns..

[Link](http://sourceforge.net/p/openhtm/todo/25/#1f4b/18dc/4963/6540/e5f8/29d4/8199/9605/f047/88f9)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

*   
  [Nick](http://sourceforge.net/u/disturbed-ua/)   
  2013-04-14

OK, I was playing with 2D and 3D visualizers and found following strange:

else if (!cell.IsPredicting && cell.WasPredicted)

{

cell.ApplySegmentUpdates(false);

}

So, take a look at the screenshot and the code. To me it seems that future prediction is taken, compared to current input and considered false.

Upd: forget it, Cells are painted red on the condition !cell.IsActive instead of !cell.IsPredicting   
and overall vizualizer colormap needs to be revised

Last edit: Nick 2013-04-15

**Attachments**

Top of Form

[](http://sourceforge.net/p/openhtm/todo/_discuss/thread/52260fdf/ba17/attachment/scr.png)  
[scr.png](http://sourceforge.net/p/openhtm/todo/_discuss/thread/52260fdf/ba17/attachment/scr.png) 

Bottom of Form

[Link](http://sourceforge.net/p/openhtm/todo/25/#ba17)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

*   
  [Itay](http://sourceforge.net/u/itay2541/)   
  2013-05-31
  + **status**: closed --> accepted

[Link](http://sourceforge.net/p/openhtm/todo/25/#287a)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

*   
  [Itay](http://sourceforge.net/u/itay2541/)   
  2013-05-31

Wait, why was this closed?  
The temporal pooler isn't working still.  
I'm waiting for a response from Michael Ferrier regarding his code.  
I am also conducting experiments in parallel.   
This issue still exists and without this getting fixed we can't continue..

[Link](http://sourceforge.net/p/openhtm/todo/25/#3b34)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* +   
    [Nick](http://sourceforge.net/u/disturbed-ua/)   
    2013-05-31

Oh..sorry. I thought we dealt with it, I saw the video where the issue is minimized with spatial pooling and on mailing list Hawkins writes that they prefer using SP.  
Michael Ferrier:

The bouncing ball example is a particularly tough one because there is so much overlap between the different frames, especially in the parts of the animation where the ball is moving slowly. One way to make this example much easier would be to do what the human visual system does; the retina just sends the brain signals for those places where a dark area is next to a light area, so the brain would get just the outline of the ball, rather than the whole filled in ball. That alone would result in much less overlap and make this example much easier for HTM.

Doug King wrote:

Hi Uwe, I think you are correct that letter bitmap sequences are an input that is not well suited to htm. A continuous motion like the bouncing ball is a more 'natural' input.

So I thought it's just our tests are incorrect.  
One thing I find strange after watching new vids, they say that similar inputs have more overlapping columns (yes, semantic encoding), in this case the issue would remain to some degree.

Last edit: Nick 2013-05-31

[Link](http://sourceforge.net/p/openhtm/todo/25/#3b34/a498)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + -   
      [Itay](http://sourceforge.net/u/itay2541/)   
      2013-05-31

Perhaps testing the temporal pooler with the jumping ball or AAAX pattern in a hard-coded region is incorrect.  
However, I think that the examples with one bit active, and the "ABBCBBA\_20x20" or "ABBCBB\_20x20" are correct.

Furthermore, I think that the "AAAX" and the bouncing ball with spatial pooler are correct. Our temporal pooler can't learn these examples at all.  
I have made a new version of the "experimental" which is working well with hardcoded as well as with spatial pooled regions.  
Also, Michael Ferrier said that our spatial pooler is not performing well.  
Also, I feel we have literally came to a halt. Where is Doug, Barry and other members to investigate the temporal pooler issue. It's now possible to debug this issue further using the visualizer. Perhaps they are waiting that Numenta will release their version? there is not much left to wait.

[Link](http://sourceforge.net/p/openhtm/todo/25/#3b34/a498/1aa6)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + - *   
        [Nick](http://sourceforge.net/u/disturbed-ua/)   
        2013-05-31

AAAX is correct but if implemented the way ABBCBB is - with no overlap.  
Have you tried bouncing ball with SP? Did you use default region settings from te project?

Also, I feel we have literally came to a halt.

They might have other things to do now and yes, we should 'synchronize' with Numenta as they release NuPic

[Link](http://sourceforge.net/p/openhtm/todo/25/#3b34/a498/1aa6/5856)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + - *   
        [Michael Ferrier](http://sourceforge.net/u/logopolis/)   
        2013-05-31

I think fixing boosting so that the patterns representing different inputs share few active columns with one another, will make a big difference for problems such as the bouncing ball.

As for the one-bit-active examples, there was some discussion of that here:

<https://sourceforge.net/p/openhtm/discussion/htm/thread/ccedad1f/#581c/74a7/00e9/1b1e/1e67>

In the cases I tested where there are two columns per input pattern, and 2 cells per column, I think it fails to represent 4 different contexts because each column chooses its learning cell independently of other columns, so there's no way to guarantee that the 2 columns of 2 cells each will choose all 4 unique combinations to represent the 4 different contexts. I haven't tested it, but I think it should work if there were 4 cells per column, in which case it should work even if there's just 1 column (of 4 cells) per input instead of 2.

[Link](http://sourceforge.net/p/openhtm/todo/25/#3b34/a498/1aa6/0662)

If you would like to refer to this comment somewhere else in this project, copy and paste the following link: 

* + - *   
        [Doug King](http://sourceforge.net/u/dwking/)   
        2013-05-31

Hi team,

It is an interesting time with Numenta activity and also where we are with openHTM. I think what has been done so far with openHTM by Itay and the rest of the team is very valuable. I have been not much help with the spatial pooler and the context forking issue - I was hoping a solution was close enough that I should get out of the way, but we seem to be stuck. However, just in the last few hours Michael Ferrier has posted about solutions to context forking on this board and also boosting, and he has open sourced his C++ version with boosting and context forking solutions implemented.

I think we may be able to move forward again. There seems to be much work ahead not just for us but Numenta too in improving the CLA, creating sharable APIs for data and creating a good user interface, implementing performance enhancements, etc.

I am confident that the current knowledge on this team as to the fundamentals of CLA is not too far form where Numenta is, and that as a team we should not abandon what we have done. It remains to be seen what Numenta has and how that project will be run, and what the dynamics of that open source project will be. We may be better off here if we have a good vision for openHTM.

I think we will be in a good position to influence the technology with what we do here, but we will need to deal with openHTMs future when Numenta puts up their project: do we continue? what are our goals and what do we want to get out of this project?

In the mean time, lets add the improvements that Michael has demonstrated and see where that gets us.