New Ideas About Temporal Pooling

gerard-leijdekkers edited this page on Jun 2 · [5 revisions](https://github.com/numenta/nupic/wiki/New-Ideas-About-Temporal-Pooling/_history)

<https://github.com/numenta/nupic/wiki/New-Ideas-About-Temporal-Pooling>

**Pages 161**



* [**Home**](https://github.com/numenta/nupic/wiki)
* [**2013 Hackathon NLP Planning Meeting Notes**](https://github.com/numenta/nupic/wiki/2013-Hackathon-NLP-Planning-Meeting-Notes)
* [**2014 Fall Hackathon**](https://github.com/numenta/nupic/wiki/2014-Fall-Hackathon)
* [**2014 Goals For NuPIC**](https://github.com/numenta/nupic/wiki/2014-Goals-For-NuPIC)
* [**2014 January Core API Meeting Notes**](https://github.com/numenta/nupic/wiki/2014-January-Core-API-Meeting-Notes)
* [**2014 Spring Hackathon Planning Meeting**](https://github.com/numenta/nupic/wiki/2014-Spring-Hackathon-Planning-Meeting)
* [**Agile Planning**](https://github.com/numenta/nupic/wiki/Agile-Planning)
* [**Anomalies**](https://github.com/numenta/nupic/wiki/Anomalies)
* [**Anomaly Detection and Anomaly Scores**](https://github.com/numenta/nupic/wiki/Anomaly-Detection-and-Anomaly-Scores)
* [**AppVeyor CI**](https://github.com/numenta/nupic/wiki/AppVeyor-CI)
* [**Build and Installation FAQ**](https://github.com/numenta/nupic/wiki/Build-and-Installation-FAQ)
* [**C 11 14 References**](https://github.com/numenta/nupic/wiki/C--11-14-References)
* [**C Coding Guide**](https://github.com/numenta/nupic/wiki/C-Coding-Guide)
* [**CHANGELOG Guidelines**](https://github.com/numenta/nupic/wiki/CHANGELOG-Guidelines)
* [**CLA and HTM Theory**](https://github.com/numenta/nupic/wiki/CLA-and-HTM-Theory)
* [**CLA Classifier**](https://github.com/numenta/nupic/wiki/CLA-Classifier)
* [**CLA for ML AI Researchers**](https://github.com/numenta/nupic/wiki/CLA-for-ML-AI-Researchers)
* [**CLA Glossary for ML**](https://github.com/numenta/nupic/wiki/CLA-Glossary-for-ML)
* [**Classifier algorithms**](https://github.com/numenta/nupic/wiki/Classifier-algorithms)
* [**Classifiers**](https://github.com/numenta/nupic/wiki/Classifiers)
* [**CMake Style Guide**](https://github.com/numenta/nupic/wiki/CMake-Style-Guide)
* [**Community Content**](https://github.com/numenta/nupic/wiki/Community-Content)
* [**Community IRC Handles**](https://github.com/numenta/nupic/wiki/Community-IRC-Handles)
* [**Comparing NuPIC to other ML techniques**](https://github.com/numenta/nupic/wiki/Comparing-NuPIC-to-other-ML-techniques)
* [**Configuration and Build System**](https://github.com/numenta/nupic/wiki/Configuration-and-Build-System)
* [**Contributing to NuPIC**](https://github.com/numenta/nupic/wiki/Contributing-to-NuPIC)
* [**Contributing to the Wiki**](https://github.com/numenta/nupic/wiki/Contributing-to-the-Wiki)
* [**Contribution standards**](https://github.com/numenta/nupic/wiki/Contribution-standards)
* [**Contributor Model**](https://github.com/numenta/nupic/wiki/Contributor-Model)
* [**Cortical Learning Algorithm**](https://github.com/numenta/nupic/wiki/Cortical-Learning-Algorithm)
* [**CPP Unit Tests**](https://github.com/numenta/nupic/wiki/CPP-Unit-Tests)
* [**Creating Applications**](https://github.com/numenta/nupic/wiki/Creating-Applications)
* [**Data Sets for NuPIC**](https://github.com/numenta/nupic/wiki/Data-Sets-for-NuPIC)
* [**Developer workflow**](https://github.com/numenta/nupic/wiki/Developer-workflow)
* [**Development Process**](https://github.com/numenta/nupic/wiki/Development-Process)
* [**Development Tips**](https://github.com/numenta/nupic/wiki/Development-Tips)
* [**Documentation Priority Listing**](https://github.com/numenta/nupic/wiki/Documentation-Priority-Listing)
* [**Documenting NuPIC with Doxygen**](https://github.com/numenta/nupic/wiki/Documenting-NuPIC-with-Doxygen)
* [**Editing Wiki Offline**](https://github.com/numenta/nupic/wiki/Editing-Wiki-Offline)
* [**Encoders**](https://github.com/numenta/nupic/wiki/Encoders)
* [**Epilepsy Seizure Prediction**](https://github.com/numenta/nupic/wiki/Epilepsy-Seizure-Prediction)
* [**Experiments To Run**](https://github.com/numenta/nupic/wiki/Experiments-To-Run)
* [**External Libraries**](https://github.com/numenta/nupic/wiki/External-Libraries)
* [**FAQ**](https://github.com/numenta/nupic/wiki/FAQ)
* [**Future Plans**](https://github.com/numenta/nupic/wiki/Future-Plans)
* [**Google Summer of Code 2014 Ideas List for NuPIC**](https://github.com/numenta/nupic/wiki/Google-Summer-of-Code-2014-Ideas-List-for-NuPIC)
* [**Hackathon Planning Checklist**](https://github.com/numenta/nupic/wiki/Hackathon-Planning-Checklist)
* [**Hackathon Resources 2014 Spring**](https://github.com/numenta/nupic/wiki/Hackathon-Resources-2014-Spring)
* [**Hackathon Resources 2015 Spring**](https://github.com/numenta/nupic/wiki/Hackathon-Resources-2015-Spring)
* [**Hierarchical Temporal Memory**](https://github.com/numenta/nupic/wiki/Hierarchical-Temporal-Memory)
* [**Hierarchical Temporal Memory Theory**](https://github.com/numenta/nupic/wiki/Hierarchical-Temporal-Memory-Theory)
* [**HTM Blogs**](https://github.com/numenta/nupic/wiki/HTM-Blogs)
* [**HTM Challenge**](https://github.com/numenta/nupic/wiki/HTM-Challenge)
* [**HTM Challenge Community Meetup Planning**](https://github.com/numenta/nupic/wiki/HTM-Challenge-Community-Meetup-Planning)
* [**HTM Challenge FAQ**](https://github.com/numenta/nupic/wiki/HTM-Challenge-FAQ)
* [**HTM Challenge Resources**](https://github.com/numenta/nupic/wiki/HTM-Challenge-Resources)
* [**HTM History**](https://github.com/numenta/nupic/wiki/HTM-History)
* [**HTM Ports**](https://github.com/numenta/nupic/wiki/HTM-Ports)
* [**Inference Types**](https://github.com/numenta/nupic/wiki/Inference-Types)
* [**Initial Season of NuPIC 2014 Planning**](https://github.com/numenta/nupic/wiki/Initial-Season-of-NuPIC-2014-Planning)
* [**Install NuPIC using Docker**](https://github.com/numenta/nupic/wiki/Install-NuPIC-using-Docker)
* [**Installing and Building NuPIC**](https://github.com/numenta/nupic/wiki/Installing-and-Building-NuPIC)
* [**Installing NuPIC on Ubuntu**](https://github.com/numenta/nupic/wiki/Installing-NuPIC-on-Ubuntu)
* [**Installing NuPIC on Ubuntu 32bit**](https://github.com/numenta/nupic/wiki/Installing-NuPIC-on-Ubuntu-32bit)
* [**Installing With Wheels**](https://github.com/numenta/nupic/wiki/Installing-With-Wheels)
* [**Intermittently Failing Tests**](https://github.com/numenta/nupic/wiki/Intermittently-Failing-Tests)
* [**Introduction to the Algorithms**](https://github.com/numenta/nupic/wiki/Introduction-to-the-Algorithms)
* [**Issue Trackers**](https://github.com/numenta/nupic/wiki/Issue-Trackers)
* [**June 2013 Hackathon Participant Info**](https://github.com/numenta/nupic/wiki/June-2013-Hackathon-Participant-Info)
* [**Learning NuPIC**](https://github.com/numenta/nupic/wiki/Learning-NuPIC)
* [**Machine Learning Competitions**](https://github.com/numenta/nupic/wiki/Machine-Learning-Competitions)
* [**Mailing List Etiquette**](https://github.com/numenta/nupic/wiki/Mailing-List-Etiquette)
* [**Mailing List Status**](https://github.com/numenta/nupic/wiki/Mailing-List-Status)
* [**Mailman Spam Filtering**](https://github.com/numenta/nupic/wiki/Mailman-Spam-Filtering)
* [**Marked for Deletion**](https://github.com/numenta/nupic/wiki/Marked-for-Deletion)
* [**Milestone View**](https://github.com/numenta/nupic/wiki/Milestone-View)
* [**ML AI Working Page**](https://github.com/numenta/nupic/wiki/ML-AI-Working-Page)
* [**Models**](https://github.com/numenta/nupic/wiki/Models)
* [**MySQL Settings**](https://github.com/numenta/nupic/wiki/MySQL-Settings)
* [**Natural Language Processing**](https://github.com/numenta/nupic/wiki/Natural-Language-Processing)
* [**NDPR Jan 26, 2015**](https://github.com/numenta/nupic/wiki/NDPR-Jan-26,-2015)
* [**Network API**](https://github.com/numenta/nupic/wiki/Network-API)
* [**New Ideas About Temporal Pooling**](https://github.com/numenta/nupic/wiki/New-Ideas-About-Temporal-Pooling)
* [**NLP Projects**](https://github.com/numenta/nupic/wiki/NLP-Projects)
* [**NLP Tools**](https://github.com/numenta/nupic/wiki/NLP-Tools)
* [**Numenta Developer Setup**](https://github.com/numenta/nupic/wiki/Numenta-Developer-Setup)
* [**NuPIC 2014 Fall Hackathon Jeff Whiteboard Chat Questions**](https://github.com/numenta/nupic/wiki/NuPIC-2014-Fall-Hackathon-Jeff-Whiteboard-Chat-Questions)
* [**NuPIC API A bird's eye view**](https://github.com/numenta/nupic/wiki/NuPIC-API---A-bird's-eye-view)
* [**NuPIC Architecture**](https://github.com/numenta/nupic/wiki/NuPIC-Architecture)
* [**NuPIC Binary Installation on Ubuntu 14**](https://github.com/numenta/nupic/wiki/NuPIC-Binary-Installation-on-Ubuntu-14)
* [**NuPIC Community**](https://github.com/numenta/nupic/wiki/NuPIC-Community)
* [**NuPIC Community Roles**](https://github.com/numenta/nupic/wiki/NuPIC-Community-Roles)
* [**NuPIC Consumer Engagement Strategy**](https://github.com/numenta/nupic/wiki/NuPIC-Consumer-Engagement-Strategy)
* [**NuPIC Core Network API**](https://github.com/numenta/nupic/wiki/NuPIC-Core-Network-API)
* [**NuPIC Directory Structure Design**](https://github.com/numenta/nupic/wiki/NuPIC-Directory-Structure-Design)
* [**NuPIC Input Data File Format**](https://github.com/numenta/nupic/wiki/NuPIC-Input-Data-File-Format)
* [**NuPIC Office Hour Dec 2014**](https://github.com/numenta/nupic/wiki/NuPIC-Office-Hour-Dec-2014)
* [**NuPIC Office Hour Feb 2015**](https://github.com/numenta/nupic/wiki/NuPIC-Office-Hour-Feb-2015)
* [**NuPIC Office Hour Nov 2014**](https://github.com/numenta/nupic/wiki/NuPIC-Office-Hour-Nov-2014)
* [**NuPIC Office Hour Sep 2014**](https://github.com/numenta/nupic/wiki/NuPIC-Office-Hour-Sep-2014)
* [**NuPIC Repositories**](https://github.com/numenta/nupic/wiki/NuPIC-Repositories)
* [**NuPIC Usage FAQ**](https://github.com/numenta/nupic/wiki/NuPIC-Usage-FAQ)
* [**NuPIC's Dependency on nupic.core**](https://github.com/numenta/nupic/wiki/NuPIC's-Dependency-on-nupic.core)
* [**nupic.core Extraction Plan**](https://github.com/numenta/nupic/wiki/nupic.core-Extraction-Plan)
* [**Office Hours**](https://github.com/numenta/nupic/wiki/Office-Hours)
* [**Online Learning**](https://github.com/numenta/nupic/wiki/Online-Learning)
* [**Online Prediction Framework**](https://github.com/numenta/nupic/wiki/Online-Prediction-Framework)
* [**OPF Code Review (29 Aug 2011)**](https://github.com/numenta/nupic/wiki/OPF-Code-Review-(29-Aug-2011))
* [**Optimization & Profiling, Speed**](https://github.com/numenta/nupic/wiki/Optimization-&-Profiling,-Speed)
* [**Orphans**](https://github.com/numenta/nupic/wiki/Orphans)
* [**Other HTM CLA projects**](https://github.com/numenta/nupic/wiki/Other-HTM-CLA-projects)
* [**Pearls from the ML**](https://github.com/numenta/nupic/wiki/Pearls-from-the-ML)
* [**Pre Sprint Planning Tasks**](https://github.com/numenta/nupic/wiki/Pre-Sprint-Planning-Tasks)
* [**Projects**](https://github.com/numenta/nupic/wiki/Projects)
* [**Python Style Guide**](https://github.com/numenta/nupic/wiki/Python-Style-Guide)
* [**Reading List**](https://github.com/numenta/nupic/wiki/Reading-List)
* [**Reconstruction**](https://github.com/numenta/nupic/wiki/Reconstruction)
* [**Release Process**](https://github.com/numenta/nupic/wiki/Release-Process)
* [**River View and HTM**](https://github.com/numenta/nupic/wiki/River-View-and-HTM)
* [**Road Map**](https://github.com/numenta/nupic/wiki/Road-Map)
* [**Roadmap**](https://github.com/numenta/nupic/wiki/Roadmap)
* [**Roles of the Flag Bearer**](https://github.com/numenta/nupic/wiki/Roles-of-the-Flag-Bearer)
* [**Running Nupic in a Virtual Machine**](https://github.com/numenta/nupic/wiki/Running-Nupic-in-a-Virtual-Machine)
* [**Running NuPIC on Cloud Amazon EC2, Google GCE, GROK**](https://github.com/numenta/nupic/wiki/Running-NuPIC-on-Cloud---Amazon-EC2,-Google-GCE,-GROK)
* [**Running NuPIC on Crouton Ubuntu Chromebook**](https://github.com/numenta/nupic/wiki/Running-NuPIC-on-Crouton-Ubuntu-Chromebook)
* [**Running NuPIC on OS X**](https://github.com/numenta/nupic/wiki/Running-NuPIC-on-OS-X)
* [**Running NuPIC on Raspberry Pi**](https://github.com/numenta/nupic/wiki/Running-NuPIC-on-Raspberry-Pi)
* [**Running NuPIC on Windows**](https://github.com/numenta/nupic/wiki/Running-NuPIC-on-Windows)
* [**Running Swarms**](https://github.com/numenta/nupic/wiki/Running-Swarms)
* [**Season of NuPIC 2014 Idea List**](https://github.com/numenta/nupic/wiki/Season-of-NuPIC-2014-Idea-List)
* [**Season of NuPIC FAQ**](https://github.com/numenta/nupic/wiki/Season-of-NuPIC-FAQ)
* [**Sensorimotor Integration**](https://github.com/numenta/nupic/wiki/Sensorimotor-Integration)
* [**Serialization**](https://github.com/numenta/nupic/wiki/Serialization)
* [**SoN 2014 Final Report**](https://github.com/numenta/nupic/wiki/SoN-2014-Final-Report)
* [**SoN 2014 Mid Term Reports**](https://github.com/numenta/nupic/wiki/SoN-2014-Mid-Term-Reports)
* [**SoN 2014 Projects**](https://github.com/numenta/nupic/wiki/SoN-2014-Projects)
* [**Sparse Distributed Representations**](https://github.com/numenta/nupic/wiki/Sparse-Distributed-Representations)
* [**Sparse Distributed Representations Quiz**](https://github.com/numenta/nupic/wiki/Sparse-Distributed-Representations-Quiz)
* [**Spatial Classification**](https://github.com/numenta/nupic/wiki/Spatial-Classification)
* [**Spatial Pooler**](https://github.com/numenta/nupic/wiki/Spatial-Pooler)
* [**Spatial Pooling Quiz**](https://github.com/numenta/nupic/wiki/Spatial-Pooling-Quiz)
* [**Sprint 21 Plan**](https://github.com/numenta/nupic/wiki/Sprint-21-Plan)
* [**Sprint 26 Planning**](https://github.com/numenta/nupic/wiki/Sprint-26-Planning)
* [**Sprint 27 Planning**](https://github.com/numenta/nupic/wiki/Sprint-27-Planning)
* [**Sprint 28 Planning**](https://github.com/numenta/nupic/wiki/Sprint-28-Planning)
* [**Sprint 29 Planning**](https://github.com/numenta/nupic/wiki/Sprint-29-Planning)
* [**Sprint Planning Meetings**](https://github.com/numenta/nupic/wiki/Sprint-Planning-Meetings)
* [**Strategic Roadmap**](https://github.com/numenta/nupic/wiki/Strategic-Roadmap)
* [**Swarming Algorithm**](https://github.com/numenta/nupic/wiki/Swarming-Algorithm)
* [**Temporal classification**](https://github.com/numenta/nupic/wiki/Temporal-classification)
* [**Temporal Memory**](https://github.com/numenta/nupic/wiki/Temporal-Memory)
* [**Temporal Memory Quiz**](https://github.com/numenta/nupic/wiki/Temporal-Memory-Quiz)
* [**Temporal Pooler**](https://github.com/numenta/nupic/wiki/Temporal-Pooler)
* [**Temporal Pooling**](https://github.com/numenta/nupic/wiki/Temporal-Pooling)
* [**Travis CI**](https://github.com/numenta/nupic/wiki/Travis-CI)
* [**Using NuPIC**](https://github.com/numenta/nupic/wiki/Using-NuPIC)
* [**Vision Object Recognition Using NuPIC**](https://github.com/numenta/nupic/wiki/Vision-Object-Recognition-Using-NuPIC)
* [**Wiki Administration**](https://github.com/numenta/nupic/wiki/Wiki-Administration)
* [**Windows Support Group**](https://github.com/numenta/nupic/wiki/Windows-Support-Group)
* [**Zeta 1**](https://github.com/numenta/nupic/wiki/Zeta-1)
* [**Zeta 2**](https://github.com/numenta/nupic/wiki/Zeta-2)
* [Show 146 more pages…](https://github.com/numenta/nupic/wiki/New-Ideas-About-Temporal-Pooling)
* [Home](https://github.com/numenta/nupic/wiki/Home)
  + [HTM Theory](https://github.com/numenta/nupic/wiki/Hierarchical-Temporal-Memory-Theory)
  + [Installing and Building](https://github.com/numenta/nupic/wiki/Installing-and-Building-NuPIC)
  + [Using NuPIC](https://github.com/numenta/nupic/wiki/Using-NuPIC)
  + [Learning NuPIC](https://github.com/numenta/nupic/wiki/Learning-NuPIC)
  + [Creating Applications](https://github.com/numenta/nupic/wiki/Creating-Applications)
  + [Contributing to NuPIC](https://github.com/numenta/nupic/wiki/Contributing-to-NuPIC)
* [HTM Challenge](https://github.com/numenta/nupic/wiki/HTM-Challenge)
* [NuPIC Community](https://github.com/numenta/nupic/wiki/NuPIC-Community)
* [NuPIC Projects](https://github.com/numenta/nupic/wiki/Projects)
* [Future Plans](https://github.com/numenta/nupic/wiki/Future-Plans)
* [Reading List](https://github.com/numenta/nupic/wiki/Reading-List)
* [NuPIC Code Repositories](https://github.com/numenta/nupic/wiki/NuPIC-Repositories)
* [Issue Trackers](https://github.com/numenta/nupic/wiki/Issue-Trackers)
* [FAQ](https://github.com/numenta/nupic/wiki/FAQ)

**Please read** [**Contributing to the Wiki**](https://github.com/numenta/nupic/wiki/Contributing-to-the-Wiki) **before editing any wiki pages.**

**Clone this wiki locally**



[**Clone in Desktop**](https://windows.github.com/)

Current info here: [Temporal Pooling](https://github.com/numenta/nupic/wiki/Temporal-Pooling)

Discussion about this text is on the [nupic-discuss](http://lists.numenta.org/pipermail/nupic_lists.numenta.org/2014-January/002662.html) mailing list.

Temporal Pooling, or TP, was described in the HTM/CLA whitepaper. However, the mechanism I proposed for TP always had problems (both biological and theoretical). It was close but we could never get it to work cleanly. I now have a new proposal for how temporal pooling works. It is more elegant and more powerful. I have not worked through all the details yet, but several people asked to hear my current thinking on this so that is what this email is about.

When working on the new temporal pooling mechanism I had further insights that led me to a better understanding of why the cortex has multiple layers of cells and how they interact. This is a major extension of HTM theory and I will also briefly describe that. The new ideas on temporal pooling and cortical layers in this email are untested; please consider them as speculative and unproven.

First a little background. The CLA consists of three components.

1. **Spatial Pooler**: This SP converts a sparse distributed input into a new SDR with a fixed number of bits and a relatively fixed sparseness. Each bit output by the SP corresponds to a column of cells.
2. **Sequence memory**: The CLA sequence memory learns sequences of SDRs. It uses the columns to represent inputs uniquely in different contexts.
3. **Temporal Pooler**: The TP forms a stable representation over sequences.

(Unfortunately we got into the habit of using the term “temporal pooler” for both the sequence memory and temporal pooling proper. In this document temporal pooling will only refer to forming stable representations over a sequence of patterns.)

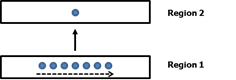
The basic idea of temporal pooling is patterns that occur adjacent in time probably have a common underlying cause and therefore the brain forms a stable representation for a series of input patterns. An example is a spoken word. If we hear a word several times we learn the sequence of sounds and then form a stable representation for the word. The input to the ears is changing but elsewhere there are cells that are stable throughout the word. Another example is when looking at an image of a familiar face. Several times a second your eyes fixate on a different part of the image causing a complete change of input. Despite this changing input stream your perception is stable. Several levels up in the cortical hierarchy there are cells that are selective for the particular person you are seeing and these cells stay active even though the input from the eyes are changing. (The most well-known of these experiments involve cells that are selective for images of celebrities, such as Jennifer Aniston. They were found while fully conscious humans had their brains exposed prior to surgery.) Of course, a single cell cannot learn to recognize an entire face. This requires a hierarchy where each level in the hierarchy is temporal pooling.

In both these cases cells remain active for multiple distinct feedforward input patterns. Cells learn to recognize and respond to different feedforward patterns when those patterns occur one after another in time. Temporal pooling is a deduced property, we can be confident it is happening throughout the cortex.

**New Idea Number 1**

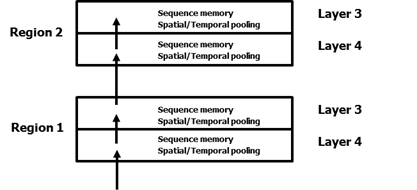
Temporal pooling occurs between layers of cells, not just between regions.

In On Intelligence I wrote that temporal pooling occurs between regions in the cortical hierarchy. As you ascend the hierarchy from region 1 to region 2, region 2 forms a stable representation of the changing patterns in region 1. Conversely, when a stable pattern in region 2 projects back to region 1 it invokes sequences of patterns in region 1.



In this diagram the dots represent SDRs and the row of dots represents a sequence of SDRs over time.

I now believe that temporal pooling is also occurring between layers of cells within a region. The canonical feedforward flow of information in a cortical region is from layer 4 to layer 3 then to layer 4 of the next higher region. Layer 4 also projects to layer 5 and then to layer 6. I believe layers 5 and 6 are using the same basic mechanism and I am making some progress in understanding them. For now I will restrict my comments to layers 4 and 3. Here is what I think is happening in layers 4 and 3.



Why have two layers of sequence memory, 4 and 3, in a region? What is the difference between layer 4 and layer 3?

A region of cortex is trying to build a predictive model of the changing input it receives. Sensory input changes because of two fundamental reasons. One is because your body and sensors move, the other is because objects in the world change on their own. For example if you are walking alone in a house then all the changes that occur on your sensors are because you are moving your eyes, head, and body. If you stood still and didn’t move your eyes there would be no changes in your sensory input. Another example is as you are look at a picture the changes occurring on your retinas are solely because your eyes are moving several times a second. The second reason sensory data can change is because objects in the world are changing on their own. For example if there were a dog walking in the house with you or if it barked it would cause changes on your sensors that were not caused by your own movement.

As a general rule, sub-cortical neurons that generate behavior have split axons. One branch generates the behavior and the other is sent to the cortex. If the cortex didn’t get copies of motor commands you couldn’t function. Every time you moved your eyes or turned your head it would appear as if the world was spinning and shifting.

I believe layer 4 models the changes in the input due to the body’s behavior. Layer 3 models the changes in the input that cannot be predicted by layer 4. I am able to show that if you take the standard CLA sequence memory and feed it both sensory data and motor commands (such as a sparse coding of a saccade’s direction and distance) it can learn to predict changes due to behavior. For example it can learn to predict what the eyes will see after a saccade (something known to occur in V1 and elsewhere). If layer 4 can successfully predict changes due to the body’s own behavior then the representations in the layer 4 sequence memory will be very sparse (one cell per column). If we TP over these changing patterns then the representation in layer 3 will be stable. If you were looking at a still image layer 4 would change with each saccade and layer 3 would be relatively stable. The representation in layer 3 would be independent of where you are fixating on the image. (Again, to form a fully stable representation of an image requires a hierarchy of regions.)

Any change that a layer cannot predict will be passed on as a change in the next layer. Any change that a layer can predict will be pooled and not result in a change in the next layer. Put another way, any change that cannot be predicted will continue up the hierarchy, first layer 4 to layer 3 then region to region. If you were looking at a walking dog, layer 4 would provide an input to layer 3 that after temporal pooling would be partially stable (the image of the dog) and partly not stable because the dog is moving. Layer 3 would try to learn the pure high order sequence of walking. All these examples require a hierarchy, but applied to simple problems you might not need a hierarchy.

Note that in layer 4 the order of patterns does not have to be repeatable. It does not have to be a high order sequence. The order and direction of saccades does not have to follow a set pattern. Similarly if I was walking through a house the order in which I do it, turning left or right, can vary. Layer 4 can handle this because it has a copy of the motor command generating the change. Layer 3 on the other hand does not get a copy of a motor command. The only way it can model the data is to look for high order sequences.

The concept of modeling changes due to our own behavior can be applied to touch, audition, and vision. It is a powerful idea that explains how we form representations of the world that are not independent of our sensor or body positions and why the world seems stable even though the patterns on our sensors are rapidly changing. It explains how you perceive objects independent of where the object is and how you are currently sensing it. For example, imagine you reach into your purse to grab a pair of eye glasses. The actual sensations on your skin are a short sequence of edges, corners, and surfaces. But you perceive the entire glasses. This is directly analogous to moving your eyes over an image.

I am not going to go through exactly how I believe layer 4 works here. I will do that another time. But it looks like that by changing the contextual information available to the standard CLA sequence memory you will get the desired result. Today NuPIC is equivalent to layer 3.

This concept of layer 4 and layer 3 modeling different aspects of an input stream requires temporal pooling between layers, not just between regions. Layer 3 has to form a stable representation (temporal pooling) of predicted changes in layer 4.

**New Idea Number 2**

Temporal Pooling can be combined with Spatial Pooling.

You might have noticed in the diagram above I labeled the first operation of a layer of cells “Spatial/Temporal Pooling”. It took me a long time to realize that a small change to the spatial pooler will allow it to do temporal pooling as well. And we need to do both spatial and temporal pooling as information moves from layer to layer and region to region. The required small change is to have the cells in a column to stay active longer so they learn to recognize multiple input patterns over time.

The standard SP does these steps.

* Receive as input a sequence of SDRs.
* Use a competitive process to learn a set of common spatial patterns in the input sequence.
* Assign a column of cells to be active for each spatial pattern in the set.
* Ensure each cell in a column learns the same feedforward response properties.

The trick to adding TP to the SP is the following. When the input to the SP was correctly predicted in the previous layer we want the cells in the column to remain active long enough to learn to respond to the next input pattern. If the input to the SP was not predicted in the previous layer then we don’t want cells in the column to remain active longer than the current input (the existing SP behavior). Temporal pooling is achieved by extending the activity of the cells in the column. But only when the input to the column was predicted in the previous layer.

It is easy to do this in software, but how might this happen with real neurons? (If you don’t care about the biology you might want to skip the next section.)

**New Idea Number 3**

Temporal pooling uses metabotropic synapses to distinguish predicted vs. non-predicted inputs.

One of the key requirements of temporal pooling is that we only want to do it when a sequence is being correctly predicted. For example, we don’t want to form a stable representation of a sequence of random transitions.

The old TP mechanism, the one in the white paper, proposed that a cell would learn to fire longer and longer in advance by predicting further back in time. In this way a cell would stay active as long as the sequence was predictable. There were several problems with this method that I couldn’t resolve.

The new proposal, described above, says that when a column of cells becomes active due to feedforward input the cells in the column will remain active for longer than normal if the feedforward input was from cells that predicted their activity. If the input cells were not predicted then the cells in the column should stay active only briefly.

There is a biological mechanism for this that fits well, but I have not been able to verify all the details. Here is the biological mechanism. Active synapses open and close ion channels, this happens rapidly on the order of a few milliseconds. The effect of an active synapse on the destination cell is short lived. However many synapses are paired with another type of receptor called a metabotropic receptor. If the metabotropic receptor is activated it will have a long duration effect on the destination cell, from 100s of milliseconds to several seconds. Metabotropic receptors can provide the means for keeping a cell active for a second or more, exactly what we need for temporal pooling.

Metabotropic receptors are not always activated. They require a short burst of action potentials to be invoked. As little as two action potentials 10msec apart are sufficient. Without that short burst the a metabotropic receptor will remain inactive. There is a lot of literature on metabotropic receptors and these properties. They are common in cortical neurons and the locations where they are not present also makes sense from a theory point of view.

Does a cell that is in a predictive state (depolarized) generate a small burst of action potentials when it first fires? That is what is required for the new TP mechanism to work. There is some evidence for this. Layer 5 cells are well known for starting with a short burst of action potentials under certain conditions. They are sometimes labeled “intrinsically bursting neurons” to reflect this. Some scientists have reported short bursts in layer 4 and layer 3 cells but not consistently so, unlike layer 5. We wouldn’t expect to see bursts in layer 4 cells in an anesthetized animal which is not generating behavior. The majority of experiments where individual cells are recorded are done with animals that are anesthetized and unable to move. Evidence for short bursts of action potentials under the correct conditions is the biggest missing piece of the new proposed TP mechanism.

The new TP mechanism, the combination of SP and TP, and the ideas for layers 4 and 3 are compelling. They explain a lot of things. So I am going to try hard to find the biological mechanisms that support these ideas. A lot of neuroscience details match up well but there are more I need to verify.

This year I want to empirically test the layer 4 ideas combined with temporal pooling. It should be possible to build a powerful hierarchical vision system. If we restrict it to spatial images and use saccades for training we could build the entire thing with a hierarchy of layer 4-only regions. Training such a system might be slow but inference should be fast. Perhaps the NuPIC community could do this in collaboration with some Grok engineers.

Jeff Hawkins, Jan 22, 2014

-Maintained by the [NuPIC Community](http://numenta.org/community.html).