Red Hat
Enterprise Linux

Hierarchical Temporal Memory

Anomaly Detection

Surya Prakash Pathak
Data Scientist
Open Services team

Manager: Michael Clifford

Red Hat

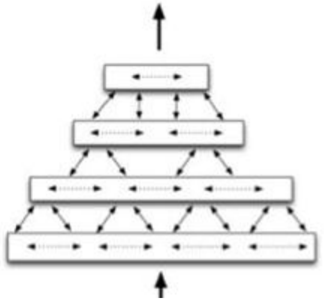
Contents

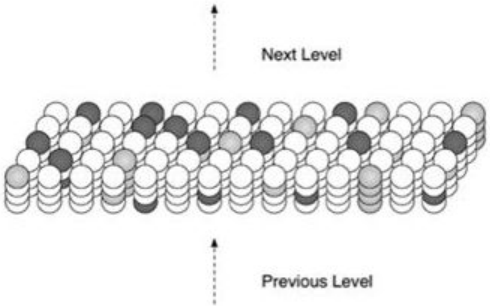
Deep Neural Networks

- Deep Neural Networks (DNN) are frameworks of Assistive Intelligence.
- Deep Neural Networks have clocked up incredible successes in many areas, (and continue to do so), however
  - DNN needs thousands if not million samples to train on.
  - DNN find it hard to adapt to continually changing data and surprises.
  - DNN are susceptible to noise and can easily be fooled. [Source]
- DNN do not work as our brain does and cannot lead to true AI.

HTM

- Hierarchical Temporal Memory (HTM) is a theoretical framework for both biological and generalised machine intelligence.
- Based on the latest understanding of the Neocortex.
- Only requires a few hundreds samples to learn.
- **Learns unsupervised** as it goes and easily handles changing data and surprises.
- Immune to up to 40% noise.
- Hierarchical (Levels of Stacked cells), Temporal (Operates over time series data), Memory (Columns of cells decides based on input, previous status of connected neighbours)
- Opens the way for truly intelligent systems.





Red Hat

History

- **2004** “On Intelligence” by Jeff Hawkins and Sandra Blakeslee.
  - The core concept in Hierarchical Temporal Memory (HTM) theory was first described in this book.

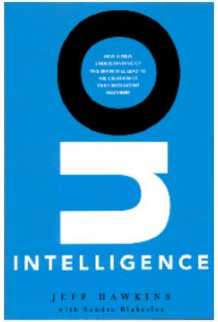
- **2005** Numenta was established in Redwood city, CA to
  - Reverse engineer the neo-cortex.
  - Apply neocortical theory to AI.

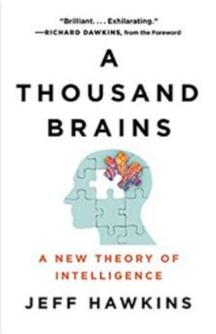
- **2014** NuPIC (Numenta Platform for Intelligent Computing) was open sourced under the AGPLv3 license.
  - API in Python 2.7, and C++

- **2015** htm.core, Community fork of nupic.core
  - API in Python 3.7+ and C++

- **2021** A Thousand Brains Theory by Jeff Hawkins
  - Further explorations of HTM and mechanism of neocortex.

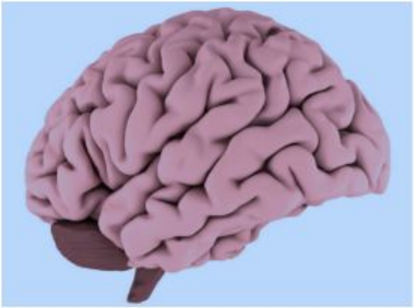
- HTM is constantly evolving with Numenta’s open research.



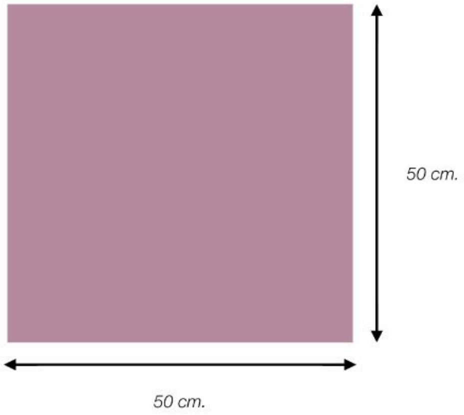


Red Hat

Neocortex

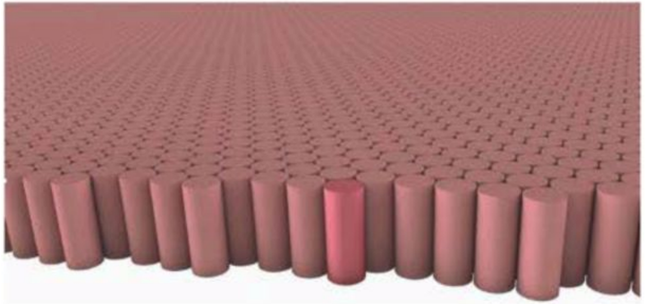


- Size of a large table napkin (50x50 sq cm)
- 75% of brains volume, 2.5 mm thick.
- 20 billion neurons, Tens of thousands of synapse per neuron.
- **Sparsely Active**
    - Only ~2% of spiking at any one time.
- Constantly predicting its inputs.
- Learns a model of the world.



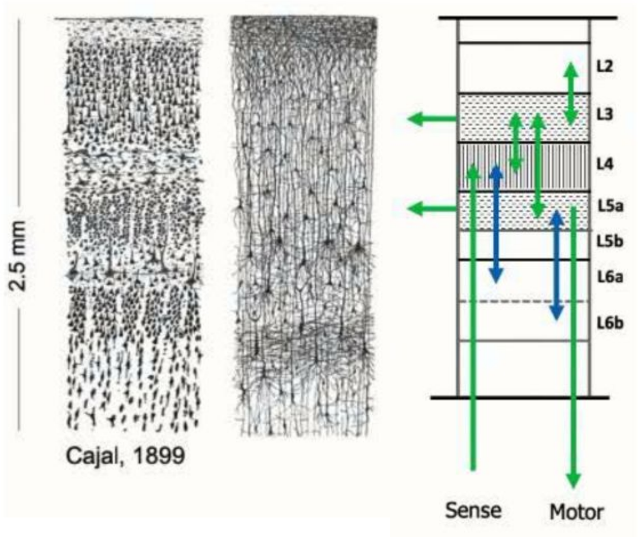
Red Hat

Neocortex Structure



- All areas in the neocortex look the same, so they must perform **same** basic function (same basic algorithm)..
- What makes one region visual and an another touch depends on what nerve they are connected to.
- The basic unit of replication is cortical column (1mm^2)
  - About 2M of them
  - So logically the cortical column is the basic unit of computation.

Cortical Column



Dozens of neuron types

Organised into layers

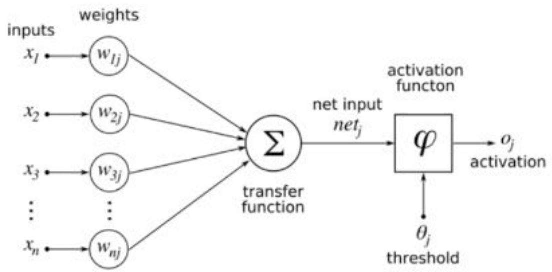
Vertical local projections cross all layers

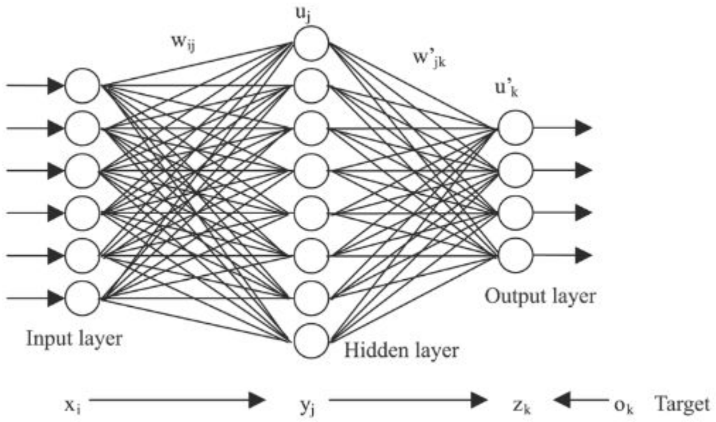
Horizontal inter-column long distance projections in some layers

- Cortical column are complex
- So whatever the column does must also be complex.
- And whatever the column does, so does the neocortex.

Deep Neural Net Neuron

- Based on the 1957 concept of the Perceptron
- Learning by adjusting the synaptic weights.
- Real neuron are not like this.

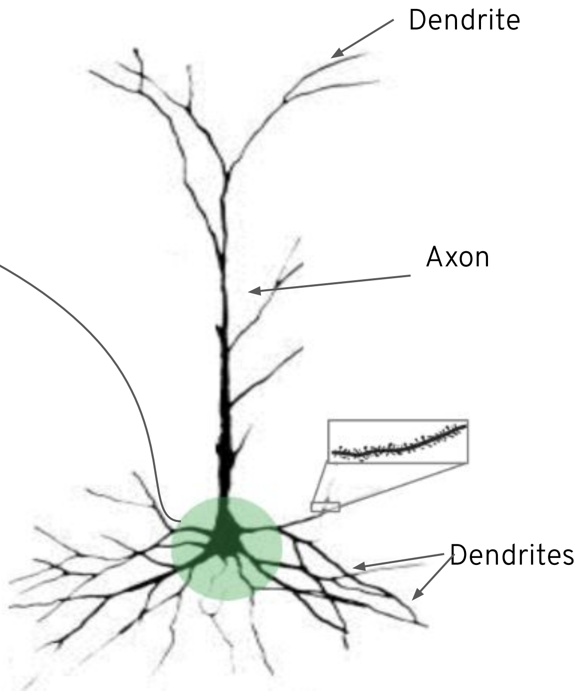




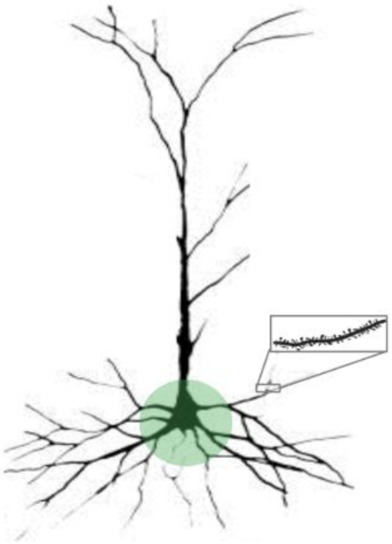
Red Hat

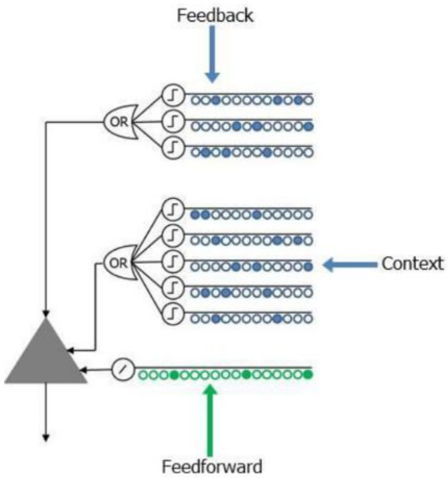
Real Neuron

- 5K to 30K excitatory synapses on the dendrites.
- 10% proximal can cause neural spike.
- 90% distal cannot cause neural spike.
- Distal dendrites are pattern detectors
    - 8-15 co-active, co-located synapses will generate a dendritic spike.
        - This puts the cell into a depolarised, or “predictive” state.
- Depolarised neurons fire sooner, inhibiting nearby neurons.



HTM Neuron

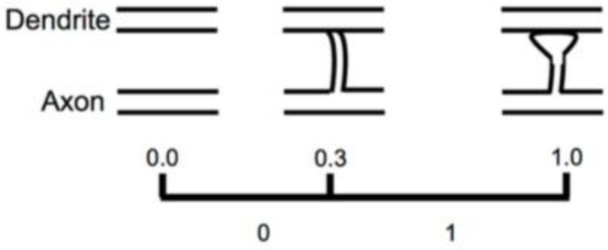




- HTM neurons don’t attempt to model all aspects of biological neurons.
- Only those that are essential for the informational aspects of the neocortex.
- HTM neuron state depends on the position and number of activated synapse – not on a sum of weights.

Red Hat

Neural Learning

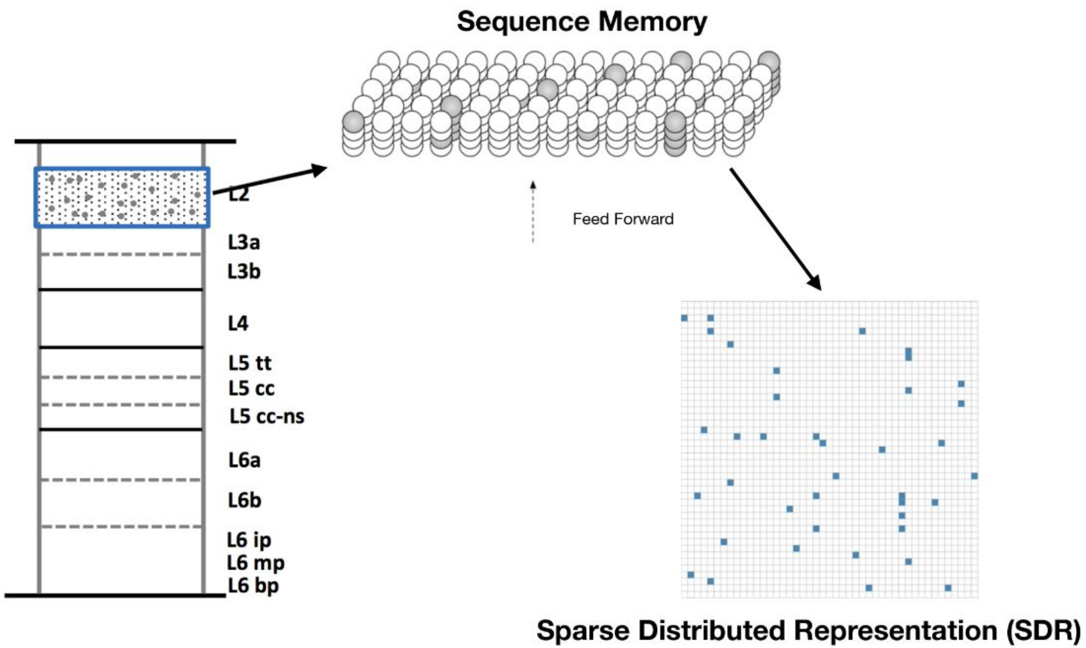


→ BIology

→ HTM - Synapse "permanence" (Synapse state)

- In HTM neurons, learning is modelled by the growth of new synapse or removal of unused synapse as in biological neurons.
- This learning occurs by incrementing or decrementing the synapse "permanence".
- A synapse is disconnected for a permanence under the threshold.
- A synapse is connected for a permanence over the threshold.
- Learning is making or breaking synapses, not adjusting synaptic weights as in DNNs.

HTM Cortical Column



Sparse Distributed Representations

- SDRs are how brains solve the problem of representing knowledge. It is used in a cortex for every aspect of cognitive function.
- Each bit has semantic meaning.
- Extremely high capacity. For 2048 bit vector and 2% are set, we have >> 10^84 unique patterns.

Capacity : $$\frac{(\text{\# OF BITS})!}{(\text{\# OF ON BITS})!\ (\text{\# OF OFF BITS})!}$$

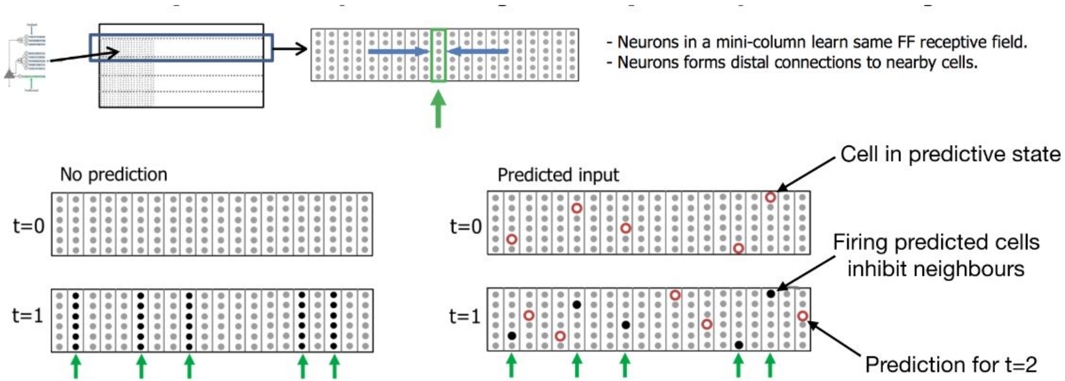
Fixed sparseness

```
010000000000000000000000100000000000000000000000000000000000000000100000..........01000
000000001000000000000000100000000000000000000000000100000000000000100000..........00000
                        ↑                                         ↑
```

- Two representations with shared bits have some shared semantic information.
- Comparing two representations is as simple as taking the intersection of the two indices sets.
- SDRs are inherently fault-tolerant and noise tolerant.

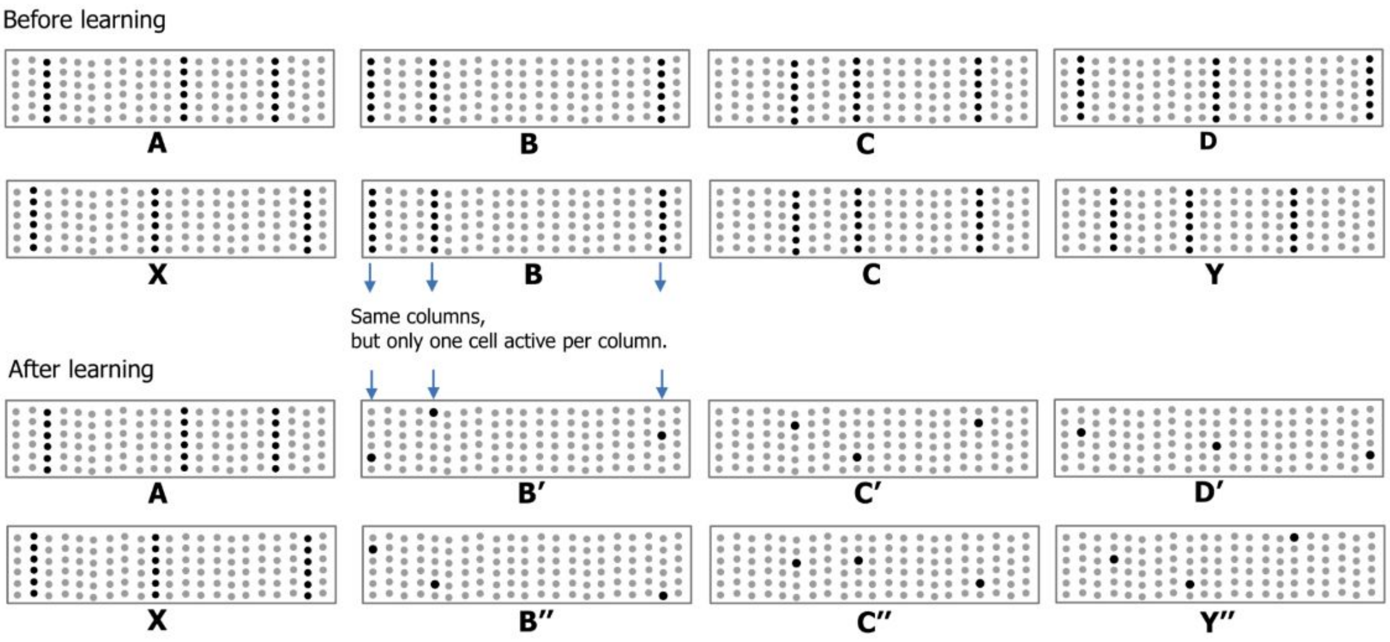
Red Hat

Sequence (formally Temporal) Memory



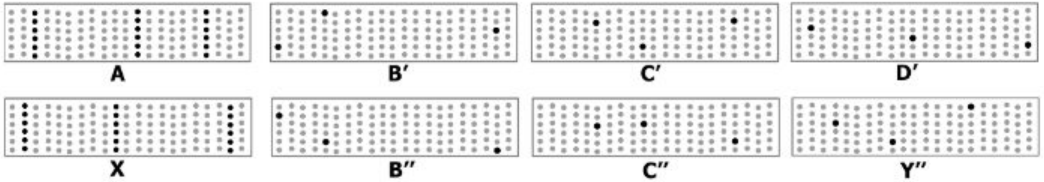
- Learns sequences of SDRs and make predictions of what the next input SDR will be.
- Extremely robust (40% noise and fault tolerant).
- Learning is unsupervised and continuous.
- Learns higher order sequences: “ABCD” vs “XBCY”.

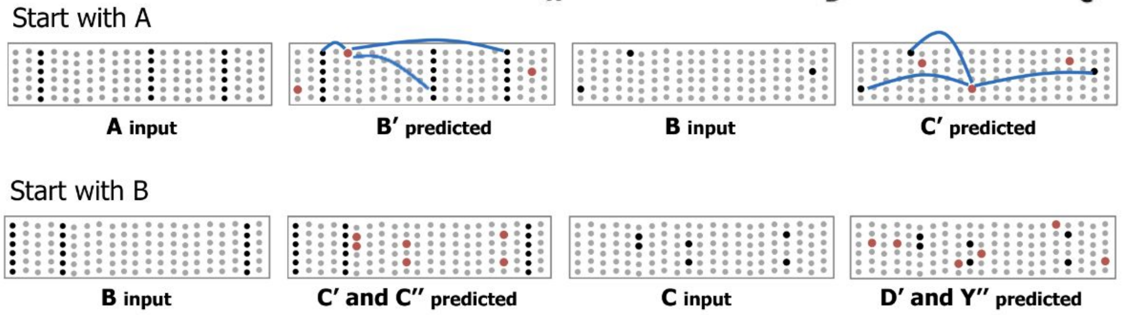
High Order Sequence Prediction



Sequence Prediction Step by Step

Trained Two Sequences A-B-C-D and X-B-C-Y



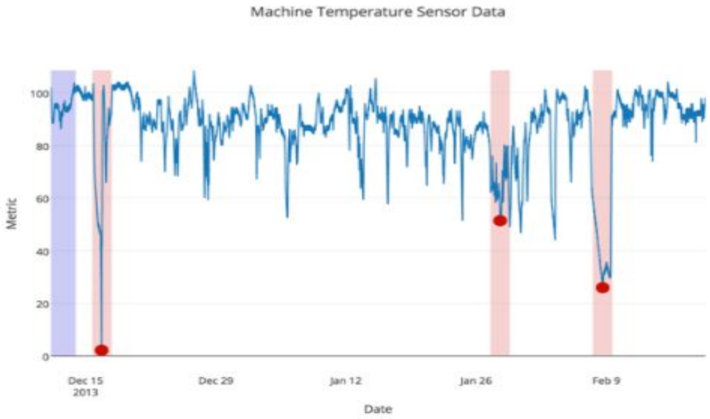


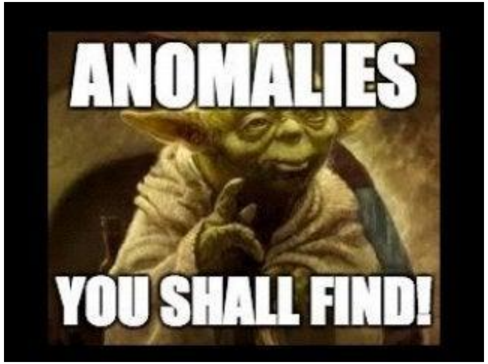
- So starting with B and inputting C predicts both D and Y.
- Hence Sequence Memory handles surprise and **multiple simultaneous predictions**.

Anomaly

What is Anomaly Detection?

- Anomalies are data points within the datasets that appears to deviate markedly from expected outputs.
- Anomaly detection refers to the problem of finding patterns in data that don't confirm to expected behavior.





Source

Red Hat

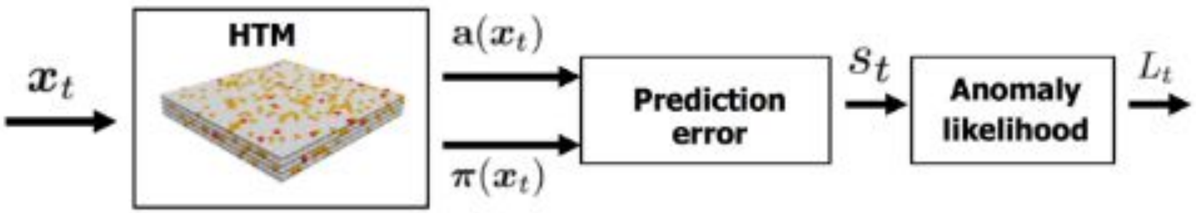
HTM- Anomaly Detection

Anomaly Score/Prediction error

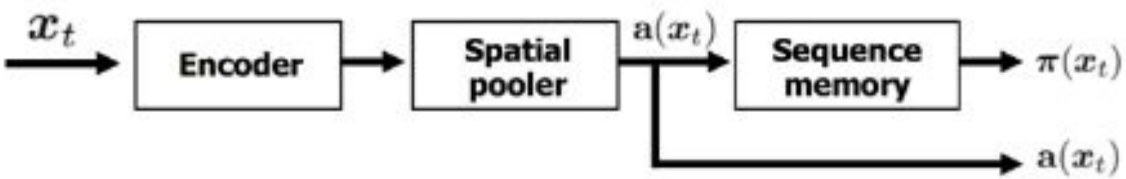
$$S_t = 1 - \frac{\pi(\mathbf{x}_{t-1}) \cdot \mathbf{a}(\mathbf{x}_t)}{|\mathbf{a}(\mathbf{x}_t)|}$$

Anomaly Likelihood

$$L_t = 1 - Q\left(\frac{\tilde{\mu}_t - \mu_t}{\sigma_t}\right)$$



HTM core algorithm components



[Source](#)

Anomaly Detection **Operate first CPU-usage data**

How data is extracted?

Thanos Programmatic Access

The Thanos metrics can be queried using the this Python client library. You can install the latest version with `pip install prometheus-api-client`. You will need to extract the OpenShift bearer token for the client library to authenticate the thanos instance.

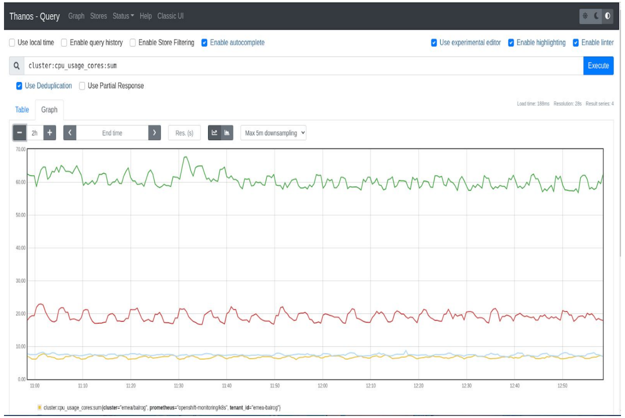
Personal token

You can use your personal token for testing and ad-hoc scripts, but it will expire in ~24 hours.

1. Login to https://console-openshift-console.apps.smaug.na.operate-first.cloud/ using `operate-first` login
2. Copy the token from the User dropdown menu
3. Use this token as your oauth bearer token when connecting to Thanos via the client library
   - Eg:

```
prometheus_api_client.prometheus_connect.PrometheusConnect(
  url='https://thanos-query-frontend-opf-observatorium.apps.smaug.na.operate-first.cloud',
  headers= {“Authorization”: “Bearer my_oauth_token_to_the_host”},
  disable_ssl=False
)
```

4. The metric data can be extracted locally using the prometheus client in a Jupyter notebook
   - Sample notebook
5. You can also spawn and run your notebooks on JupyterHub



GitHub

Red Hat

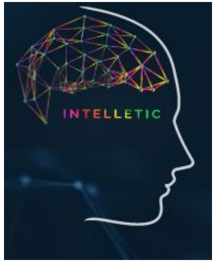
Anomaly Detection **using HTM**

Demo – Run through the notebook showing the time series data for input signal, prediction and detection of anomalies.

Commercials applications **HTM**

https://grokstream.com/

https://www.cortical.io/



https://intelletic.com/

Acknowledgements

Numenta

https://numenta.com/

htm-community / **htm.core** Public

forked from numenta/nupic.core

https://github.com/htm-community
/htm.core

★ **Special thanks to Marcel Hild**

Who am I?

- Data Scientist, Data Science Open Services team
- Currently based at Berkeley, CA.
- supathak@redhat.com
- https://www.linkedin.com/in/sppathak/
- https://github.com/suppathak

Red Hat