Hierarchical Temporal Memory (HTM)

Computational Cognitive Neuroscience
COSC 521, Spring 2019
Corey Johnson
Resources for information on HTM

Numenta.org

HTM school

NuPIC

Jeff Hawkins 2004 book:

*On Intelligence: How a New Understanding of the Brain will Lead to the Creation of Truly Intelligent Machines*

(Note: some slide graphics are from Numenta.org, HTM school)
Hierarchical
- Higher levels → abstraction & permanence

Temporal
- Change over time: patterns

Memory
- Sparse Distributed Representation (SDR)
HTMs

- Biologically plausible model for intelligence
- Based on pyramidal neurons
- Neocortex micro-columns as building blocks

"The neocortex comprises about 75% of the volume of the human brain and it is the seat of most of what we think of as intelligence." - Jeff Hawkins
HTMs continued

- Can learn / recall / infer high-order sequences
- Local learning rules, no global supervisor (HW)
- Relies on sparse distributed representation (SDR)
  - Fault tolerance
  - High capacity
- Learns by modeling the growth of new synapses
HTM Layers

(Lect. 2 slides)
Sparse Distributed Representation

Fundamental to HTM systems

SDR: 'Language of intelligence'

Sparse as opposed to dense binary code, many bits are needed

SDR bit similarity $\rightarrow$ similar semantic meaning

Semantic error is key to generalization

Cell only needs a few connections to neighbor to match pattern
Applications

Good for:

- Data streams that change over time: text, GPS, dates, numbers
- Data with inherent structure
- System where many models are required rather than one large one
- Prediction
- Anomaly detection
- Classification
HTM theory is still evolving.
Welcome to Numenta.org, home of Numenta's HTM community and open source projects. If you want to learn about Numenta the company visit Numenta.com.

Machine Intelligence Starts Here

Hierarchical Temporal Memory is a foundational technology for the future of machine intelligence based upon the biology of the neocortex. Because Numenta is committed to making this technology accessible to everyone, all HTM software and ongoing research is open source. This allows you to work with our technology in whatever way works best for you — learn about the theory, dive into the source code, or start your own implementation. Some of our community members have written their own versions of HTM systems in other languages and platforms. Others have created detailed visualizations, experiments, and applications.

The neocortex is a logical system that we'll understand fully in time. HTM theory reflects our current understanding of how the neocortex works, and HTM code reduces that theory to practice. HTM is continually being updated as we learn more about the brain. We believe HTM will play a critical role in the creation of truly intelligent machines.
HTM School videos
Implementations

Hierarchical Temporal Memory (HTM) is a theory of intelligence that can be implemented in most computer programming languages. Below are descriptions of several HTM implementations currently active within our community. For detailed descriptions of HTM algorithms, see our living text, Biological and Machine Intelligence. It contains pseudocode for both the spatial pooling and temporal memory algorithms (PDFs).

NuPIC, or the Numenta Platform for Intelligent Computing, is an HTM implementation created by Numenta and open-sourced in June 2013. This codebase is the original HTM codebase, and is architected in a way that allows algorithmic experimentation in Python, but more performant versions of HTM algorithms in C++.

NuPIC Core (C++)

Our C++ codebase contains all HTM algorithms written in C++, and SWIG language bindings to Python. Language bindings to other environments should be added here.

This codebase exposes the Network API, which is the primary low-level interface for creating HTM systems.

http://github.com/numenta/nupic.core

NuPIC (Python)

The NuPIC Python codebase contains Python code implementations of HTM. Through this interface, users may specify whether their code runs Python algorithms or the faster C++ algorithms using the Python bindings provided in nupic.core.

In addition to providing Python bindings to the nupic.core Network API, this codebase also includes a higher-level client API called the Online Prediction Framework (OPF), which is tuned towards experimentation with predictions, anomaly detection, and identifying optimal model parameters (swarming).

http://github.com/numenta/nupic
HTM Studio

Find Real-Time Anomalies in your Streaming Data

HTM Studio allows you to test whether our Hierarchical Temporal Memory (HTM) algorithms will find anomalies in your data. With just one click, you can uncover anomalies other techniques cannot find in your numeric, time-series data, in minutes.
Video Link: https://youtu.be/X50GY0mdHlw
Stocks

- APA: Apache Corporation
- PCLN: Priceline Group
- NOC: Northrop Grumman
- IBM: International Business Machines
- PCG: PG&E
- AEP: American Electric Power Co
- SCHW: Charles Schwab
- DUK: Duke Energy
- MA: MasterCard Class A
- PNC: PNC Financial Services Group
- PRU: Prudential Financial

Geospatial

Rogue Behavior
Conclusion

HTM aims to be a biologically-constrained theory of intelligence

You may hear more about HTMs in the future

 Needs killer app moment like DCNN (ImageNet) or DRL (Go/Chess)

Resources:

- Numenta.org
- HTM School
- NuPIC code
- HTM Studio
Questions / Comments / Thoughts?

Thanks!

(For more information on HTMs, please see Numenta.org)