Hierarchical Temporal Memory (HTM)

Hierarchical Temporal Memory

- My name is Connor Leahy

  - Computer Science student at TUM

  - Co-organizer here

  - Put way too much on my plate

On the last episode of “Enthusiastic Science Man”…

A way forward?

- Synthetic Gradients (Not really a solution) [5]
- Evolutionary Strategies (Google, Uber etc) (Still relies on ANN “substrate”) (Special mention to Novelty Search)
- Probabilistic Programming (Josh Tenenbaum, MIT)
- Biologically inspired methods

  → Hierarchical Temporal Memory (Jeff Hawkins, Numenta) [6]

  → BioSPAUN (Chris Eliasmith, University of Waterloo) [7]

This Talk

- The neuroscience inspiration of HTM
- High level overview of HTM
  - Sparse Distributed Representations (SDRs)
  - The HTM Neuron
  - Spatial and Temporal Pooling
- Benefits of HTM
- Where to find more

Disclaimer: HTM is incomplete and highly experimental

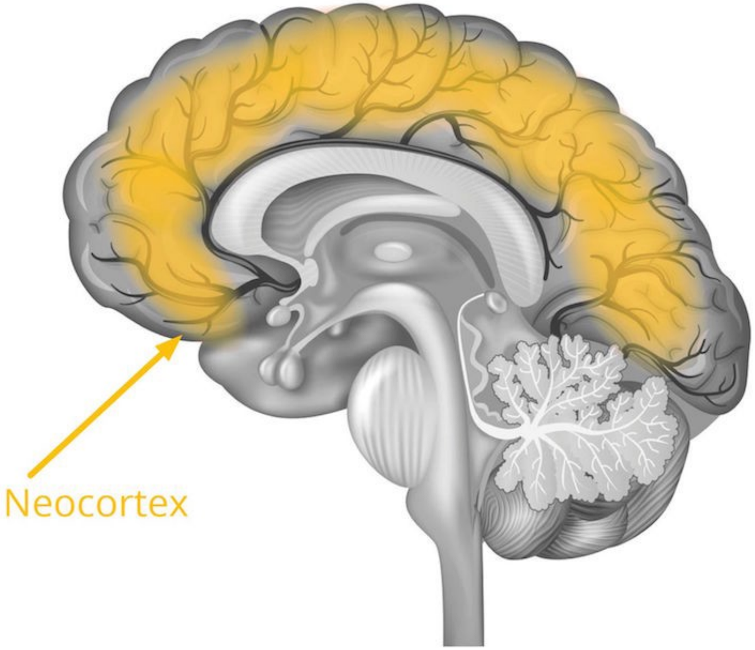
...and it's going to be impossible to go in depth, so talk to me later for the details

The Inspiration behind HTM

- Humans are kind of amazing…

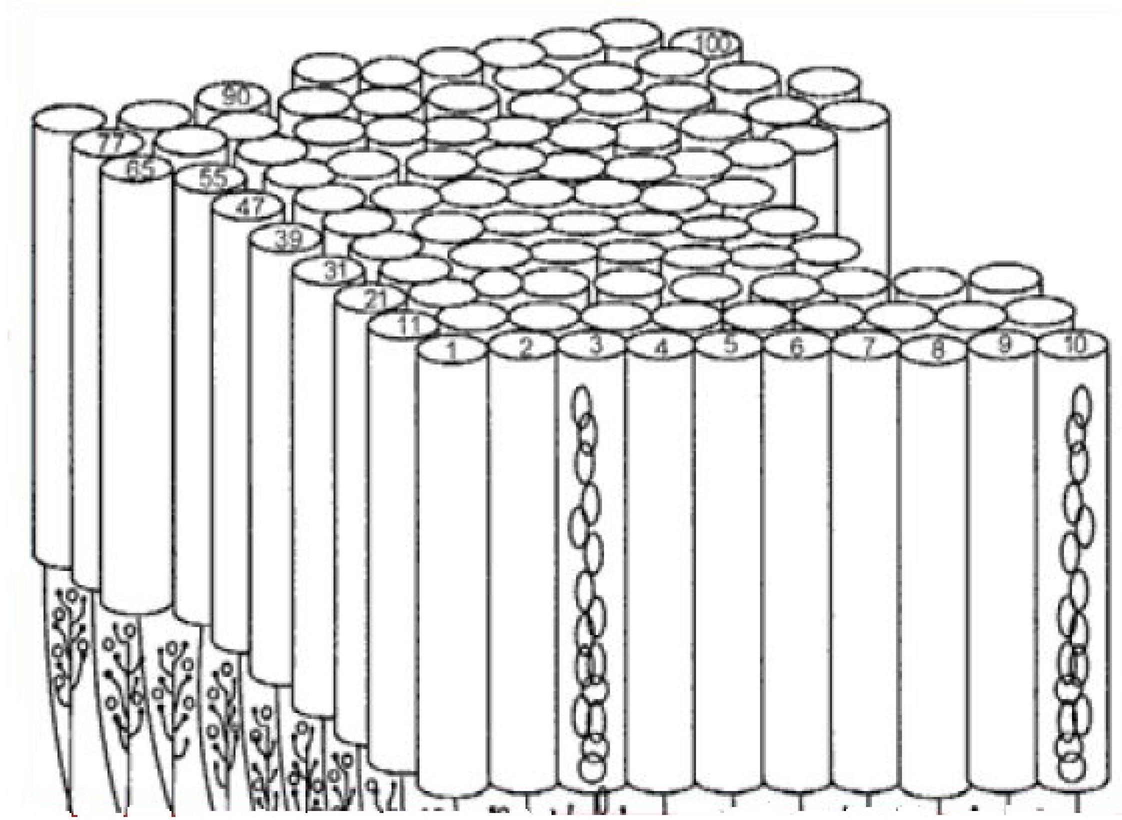
  ...so it seems they might be a good first thing to look at if we want to design intelligent machines

- HTM aims be a theory of the neocortex that strikes a balance between biological plausibility and computation usefulness



The Inspiration behind HTM

- Vernon Mountcastle’s proposal: All regions of the neocortex are doing the same thing [1]
- Proposed organization into “cortical columns”



Evidence for Mountcastle's Proposal

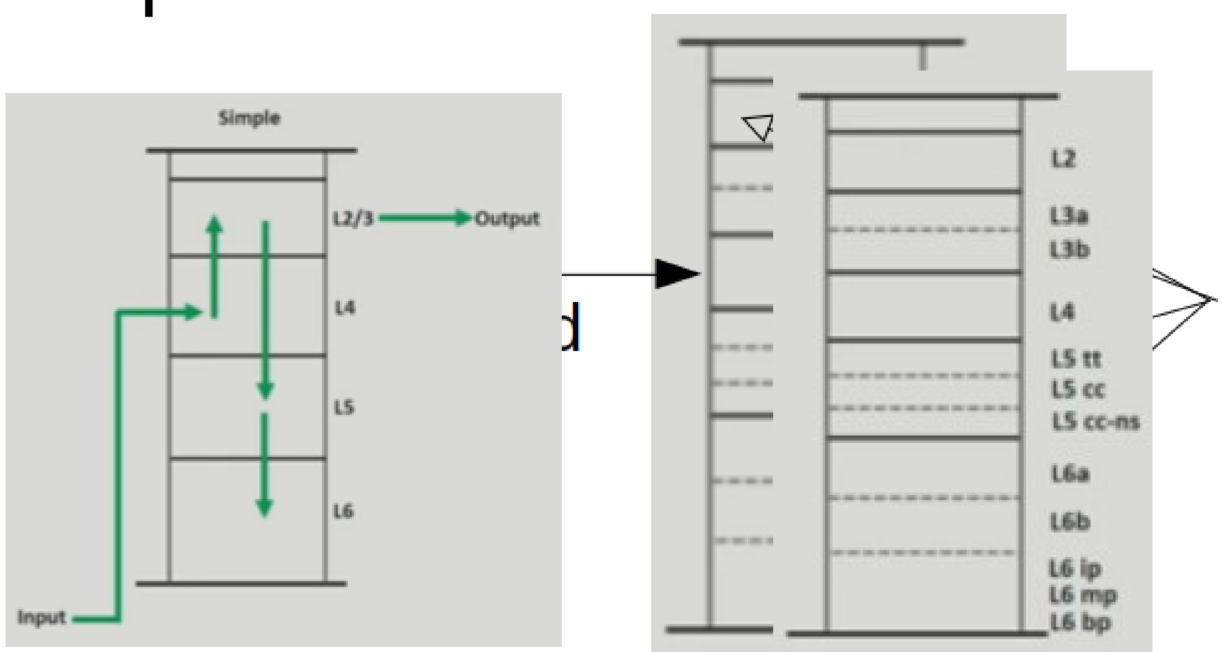
- Lots of anatomical data
- Experiment switching visual and auditory input in ferrets shows this [2]
- Most significant difference between human brains and other mammals is the size of the neocortex

High Level Overview of HTM

- The core of the theory: The neocortex predicts the future
- Memory: HTM memorizes input
- Temporal: HTM always has a sense of time, it learns sequences of inputs (You can't listen to a song all at once)
- Hierarchical: The neocortex is organized in a hierarchy

High Level Overview of HTM

- HTM is an online learning sequence memory
- At each timestep, information flows in and the network predicts its next input
- More complex functions are WIP

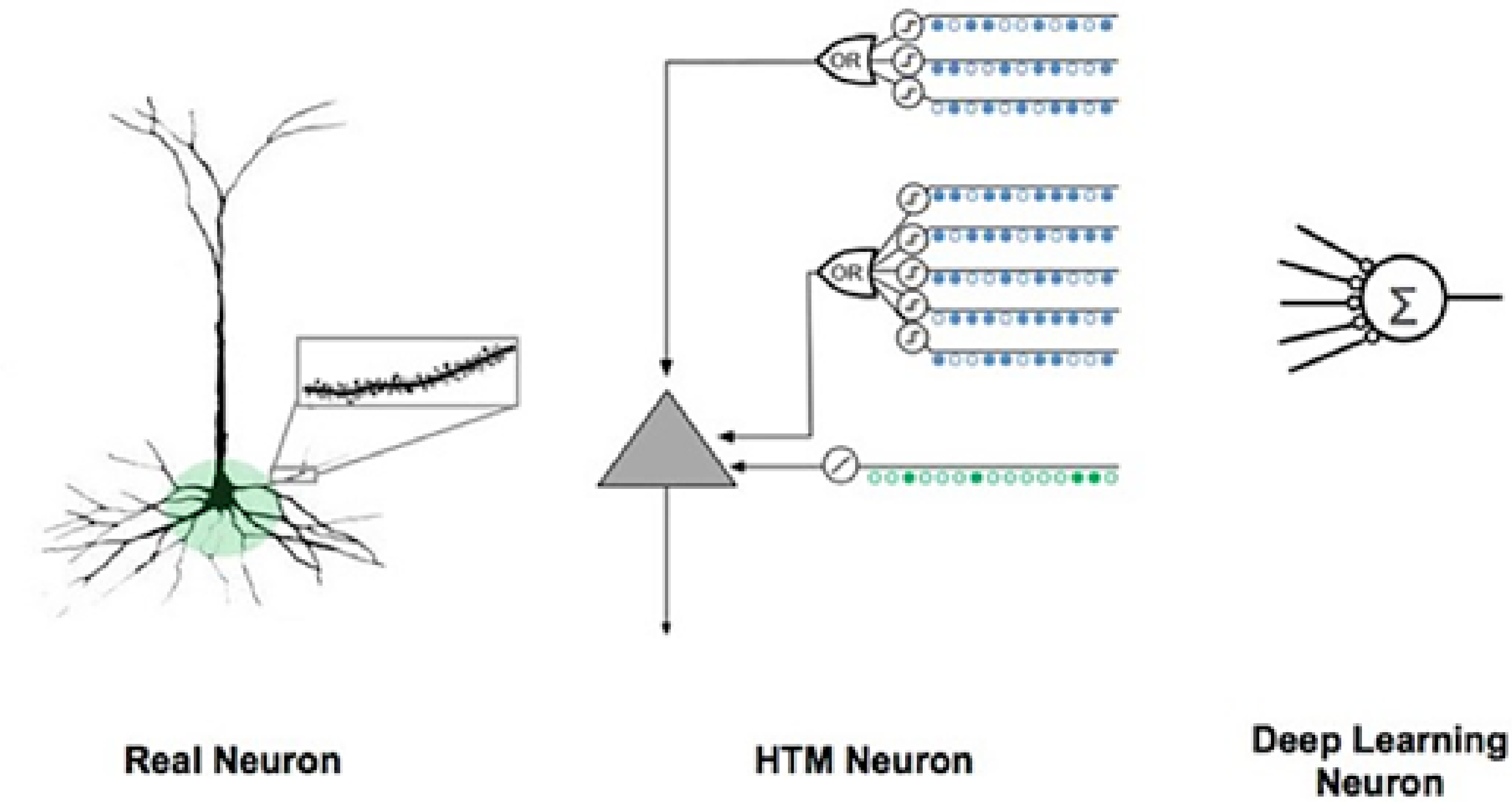


What we don't understand

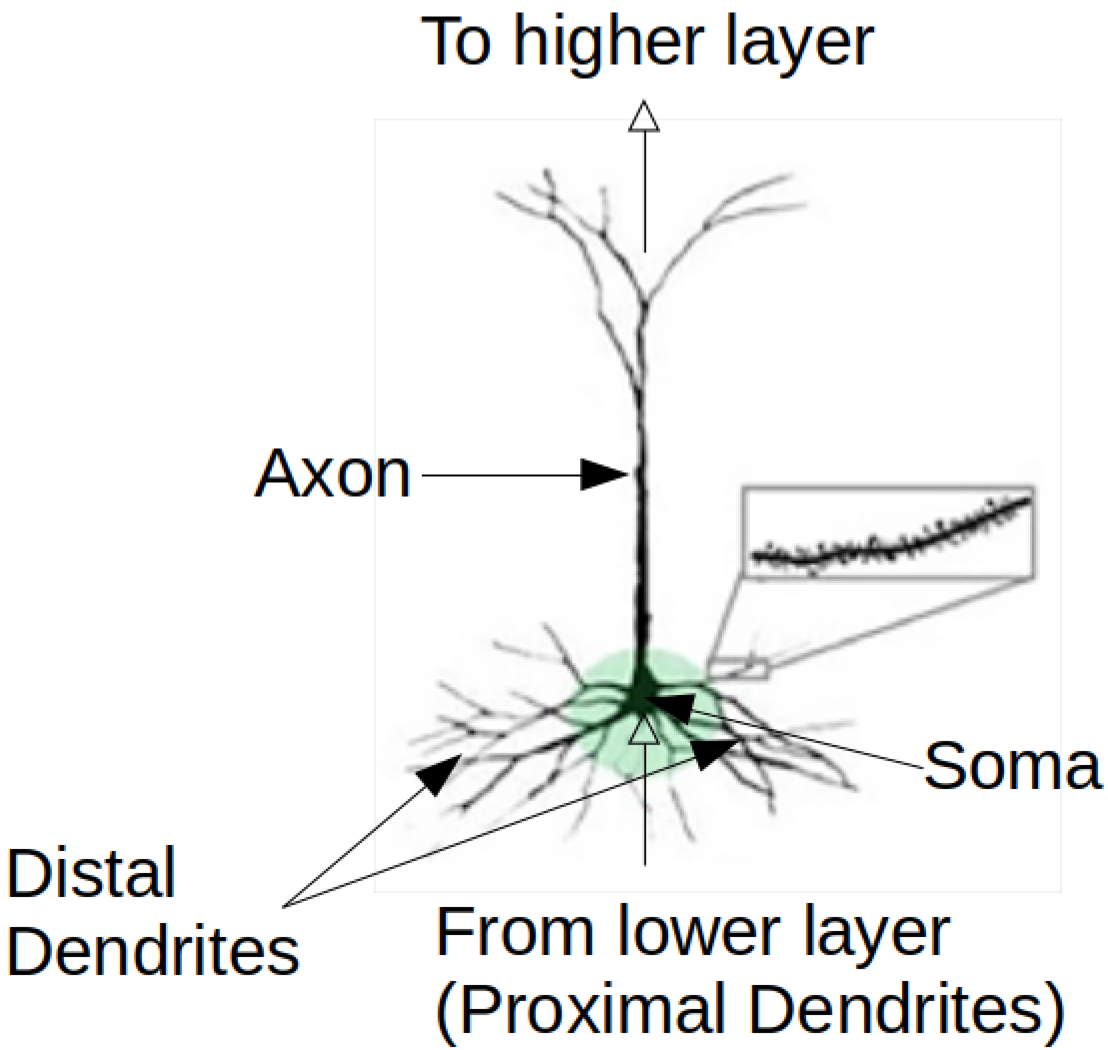
Sparse Distributed Representations (SDRs)

- The “data structure of the brain”
- Can be visualized as the on/off states of a bundle of neurons at a given timestep
- Input is encoded into SDRs (equivalent to sensory organs such as retina)

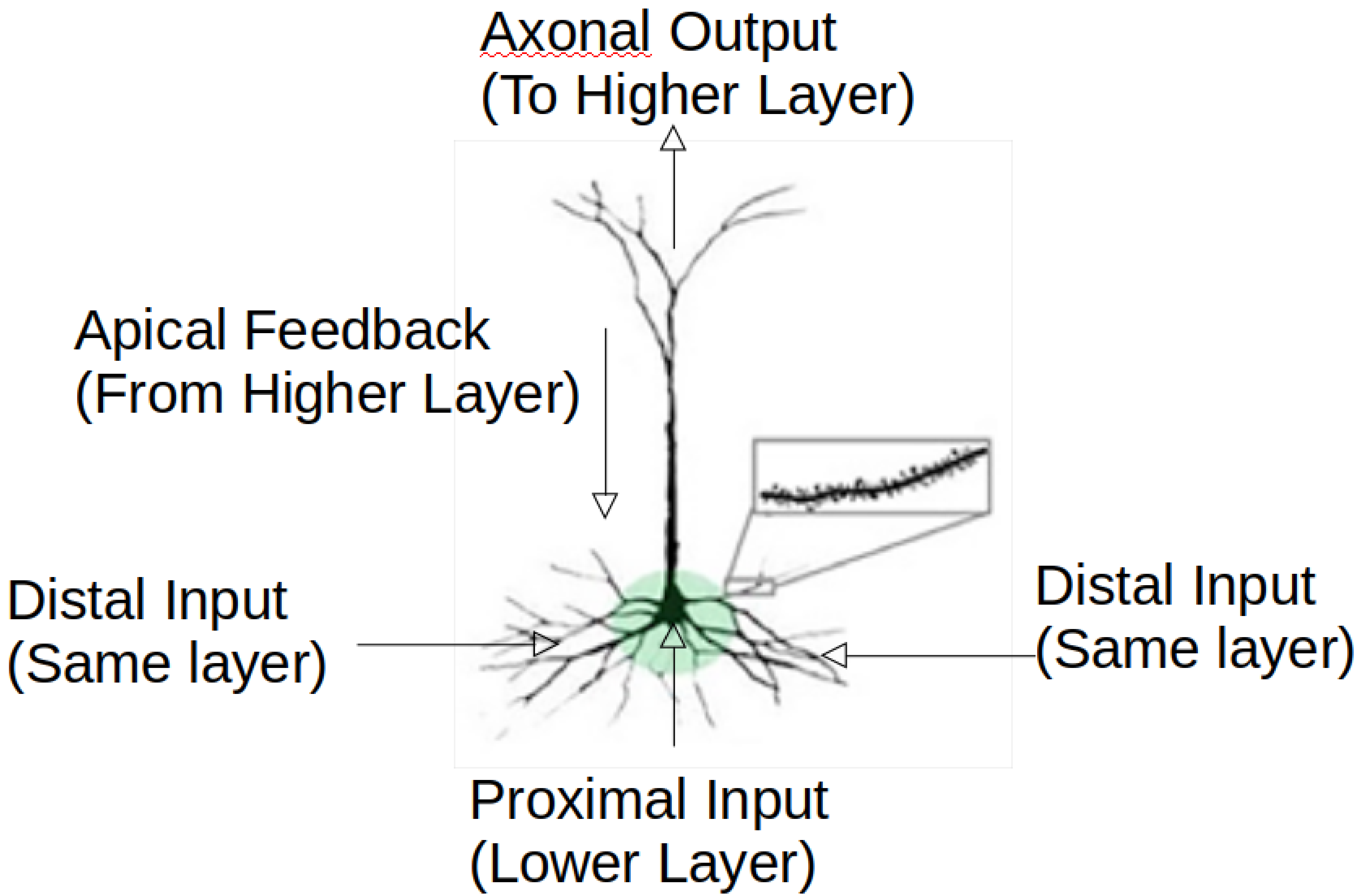
The HTM Neuron



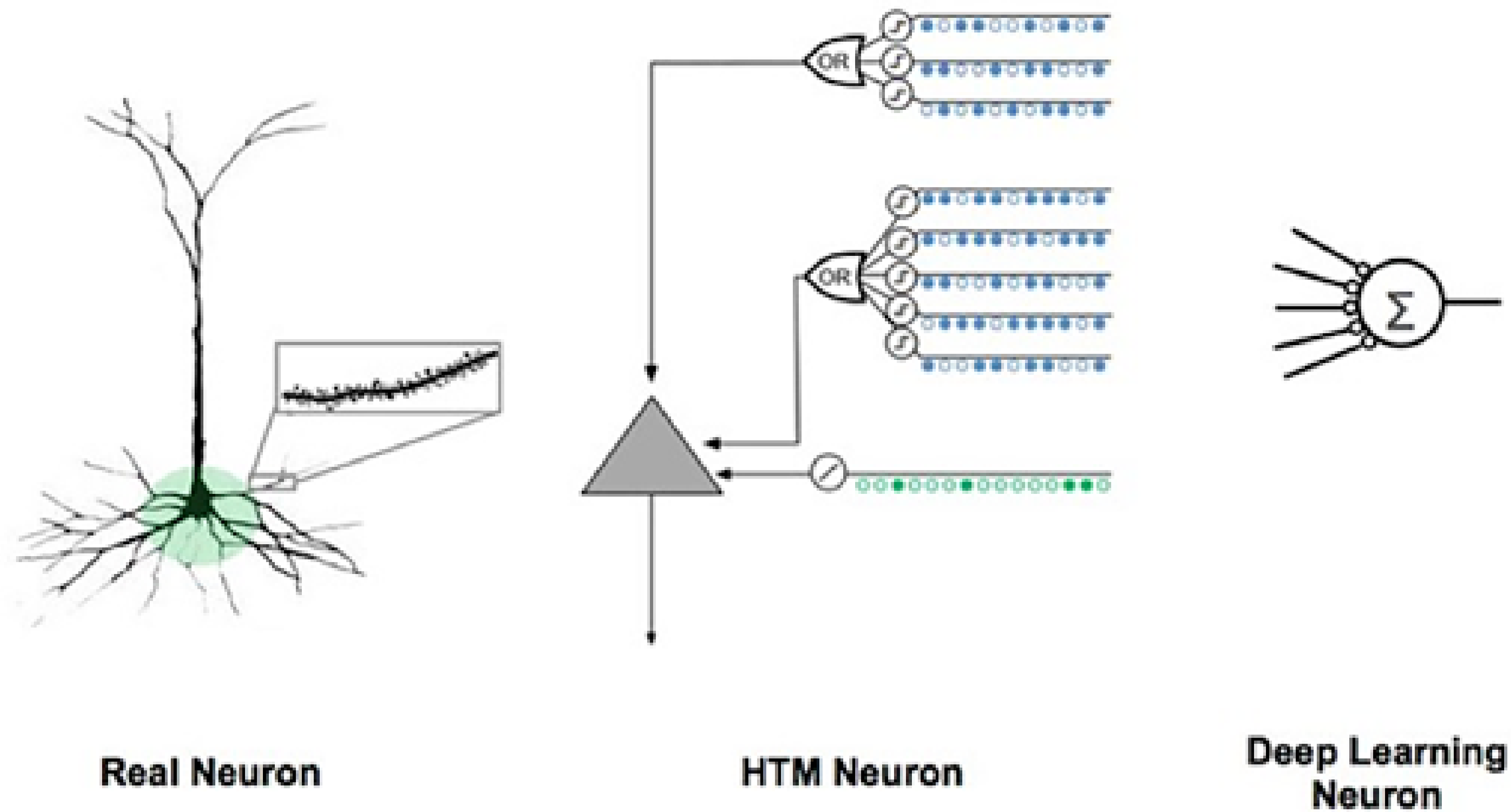
A Biological Neuron



A Biological Neuron



The HTM Neuron



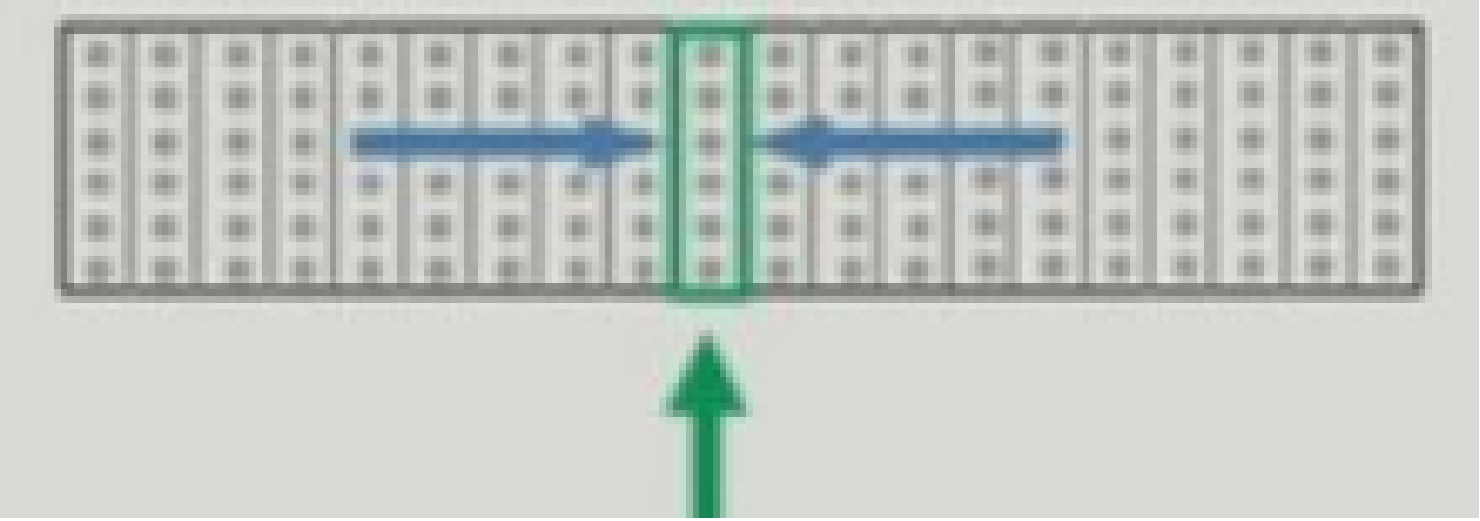
Spatial and Temporal Pooling

- Problem: We want to classify input, retain sparsity and predict the next input

  Solution: Spatial and Temporal Pooling

Spatial and Temporal Pooling

- HTM Neurons are organized into a layer of “minicolumns” (distinct from cortical columns!)
- Each minicolumn is connected to some subsample of the input space (the “receptive field”)
- Each cell has distal connections to other cells within the layer

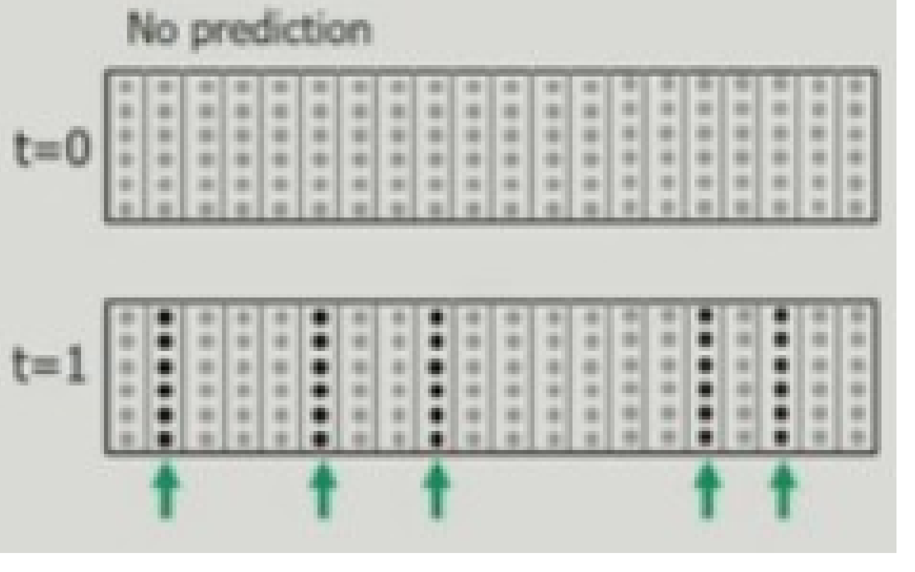


Spatial Pooling

- Every timestep:

  - A minicolumn's "overlap" is how many "on" bits it is connected to

  - The top X minicolumns with the highest overlap are set to on (leading to sparsity)



Temporal Pooling

- Spatial Pooling alone has no concept of context

  A → X → B

  C → X → D

  Assuming the network sees “X”, what should it predict?

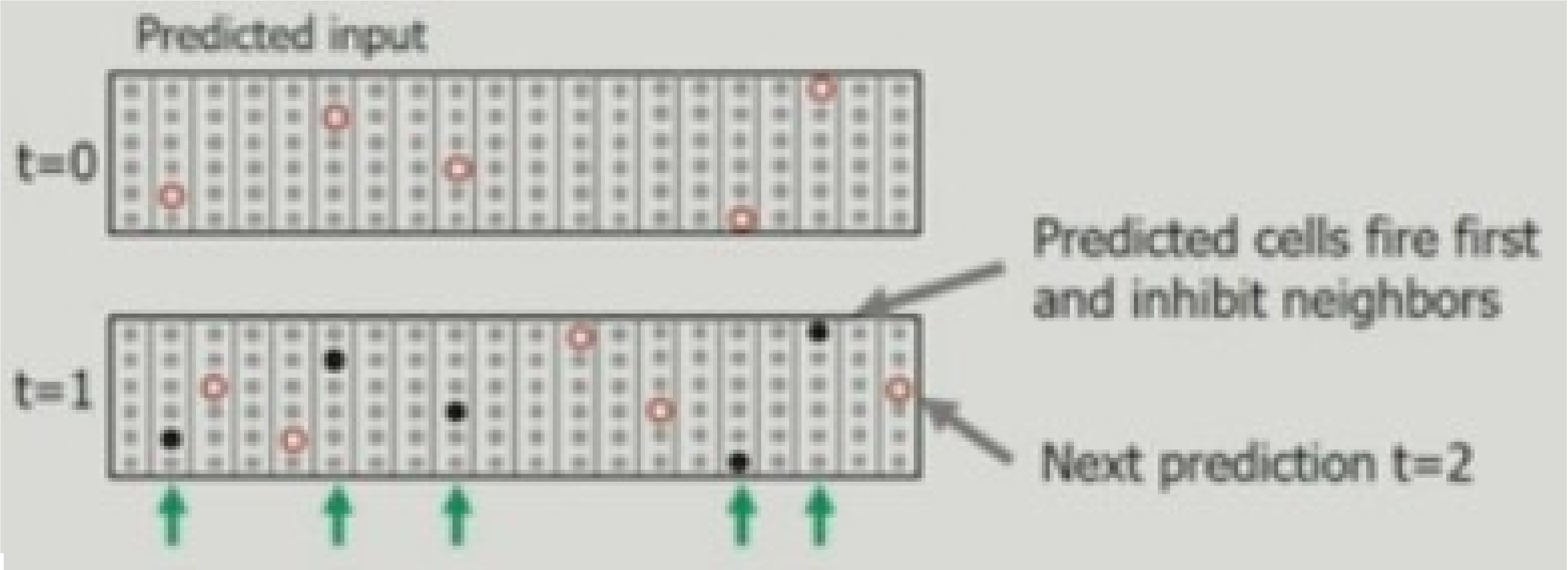
  If it previously saw “A”, it should predict “B”, similar with “C” and “D”.

  If it saw nothing before X, it predicts “B” and “D”

Temporal Pooling

- Cells in a minicolumn have varying distal connections to cells within their layer
- If a threshold of these is active in one timestep, the cell is put into a predictive state in the next timestep
- If a minicolumn fires, and one or more cells are predictive, only these fire
- If no cells are predictive, all cells in the minicolumn fire (“bursting”)

Temporal Pooling



Temporal Pooling

- The cell within a minicolumn that ends up firing defines the context

  For example, there is a minicolumn for “X”, which contains cells for “X in the context of A” and “X in the context of C”

Benefits of HTM

- There has been very little experimentation with HTM compared to DL
- HTM is not a drop in replacement for DL, but offers many new interesting ideas

Benefits of HTM

- HTM is highly fault tolerant (including graceful failure), hyper parameter insensitive, online learning, no catastrophic forgetting, unsupervised and uses only local learning rules
- Right now:

  - Sequence Memory

  - Anomaly Prediction (Grok)

  - Text Understanding (cortical.io)

Benefits of HTM

- Future:

  - Understanding of human neocortical processes

  - A step towards AGI

  - Highly energy efficient hardware (possibly far beyond von Neuman Architecture)

Where to find more

- Numenta.org (Great forum, papers and github)

  →NuPIC reference implementation
- HTM School → YouTube
- Book On Intelligence, Jeff Hawking (Somewhat outdated)
- For the new Sensorimotor stuff: Have We Missed Half of What the Neocortex Does? → Youtube

[1] Mountcastle, V. B. (1978): "An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System"

[2] Angelucci, Clasca, Bricolo, Cramer, Sur (1997): “Experimentally Induced Retinal Projections to the Ferret Auditory Thalamus: Development of Clustered Eye-Specific Patterns in a Novel Target”