Hierarchical Temporal Memory

Helmut Puhr
0230247

Computational Intelligence Seminar A
SS2009
Maass, Wolfgang, O.Univ.-Prof. Dipl.-Ing. Dr.rer.nat.
Outline

- Introduction
- On Intelligence
- Neocortex
- Hierarchical Temporal Memory
- HTM Theory
- Examples
- Criticism
- Conclusion
Introduction

Tasks

• Visual pattern recognition
• Understanding spoken language
• Recognizing and manipulating objects by touch
• Navigating in a complex world

Hard to solve using machines
“Easy” to solve using a neocortex
Introduction HTM

- „Memory“ system derived from neocortex
- Trained using (any) data
- Tree-shaped hierarchy of nodes
- Hierarchy of space and time
- Discover and infer causes
- Similar to Bayesian Networks
On Intelligence

Chinese Room (John Searle)

• Input gets manipulated according to specified rules
• Through output person may appear intelligent
• But no understanding did occur
On Understanding

New definition of intelligence

Thesis:
Understanding can't be measured by external behavior (e.g. Deep Blue etc.)

Use an internal metric of how the brain remembers things and uses its memories to make predictions
Neocortex

- Part of the brain of mammals
- Higher functions
  - Sensory perception
  - Motor commands
  - Spatial reasoning
  - Conscious thought
  - Language
Neocortex anatomy

• Gray matter surrounding white matter
• ~ 2.4 mm thick sheet
• 6 layers (different cell types, neuronal connections)
• Neurons in vertical structures
  – Neocortical columns
Neocortical columns

• Solve previously unknown problems
• Fundamentally generic algorithm
• Different problem domains

Theory of a Memory-Prediction framework
Memory-Prediction framework

• Constructs a model for the spatial and temporal patterns
• Repetitive structure of a canonical cortical circuit (node)
• Organized as a hierarchy
• Every node memorizes patterns, can predict input
• Unsupervised pattern recognition
• Information is passed up and down in the hierarchy
Memory-Prediction framework

- Biological theory
- Derive mathematical counterparts

Hierarchical Temporal Memory
Hierarchical Temporal Memory

Functions

• Discover causes in the world
• Infer causes of novel input
• Make predictions*
• Direct behavior*

* ... optional
Discover causes in the world

- “Causes” ... Objects
- Spatial, temporal data from Sensor
- For past & current input assign likelihood for causes
- Distribution of causes forms “belief”
Infer causes of novel input

- After training causes
- Perform inference (pattern recognition)
- Sensory input always novel
- Time-varying inputs
Make predictions

• Each node stores sequences of patterns
• Predict input based on stored data
• Used for
  – Priming (Noisy, incomplete, ambiguous data)
  – Imagination and Planning
Direct behaviour

- HTM attached to physical system
- Can interact with the world
- Hard-wired behavior “reflexes”
- Predict future
- Create goal-oriented behaviors
Hierarchical Temporal Memory

- Tree hierarchy of nodes
- Bottom-level nodes get sensor input
- Nodes
  - I/O
  - Similar algorithm
  - Contain memory
Why is hierarchy important?

- Shared representations lead to generalisation and storage efficiency
- The hierarchy of HTM matches the spatial and temporal hierarchy of the real world
- Belief propagation ensures all nodes quickly reach the best mutually compatible beliefs
- Hierarchical representation affords mechanism for attention
HTM Theory

Node phases

- Learning: for every input
  - Memorisation of patterns
  - Learning transition probabilities
  - Temporal grouping

- Sensing/Inference
Memorization of input patterns

• Memory stores patterns under label
• Input compared against stored patterns
• If known, identify label
• If new, store and give a label

Pattern memory matrix ... C
Rows store individual patterns
Learning transition probabilities

- Node constructs and maintains a Markov graph
- Labeled vertices correspond to stored patterns
- Edges represent normalised number of transitions
Temporal grouping

- Partition the set of vertices into a set of temporal groups
- Vertices of the same temporal group are highly connected
- Agglomerative Hierarchical Clustering
Sensing/Inference in a node

- Discard Markov graph, keep temporal groups
- Produce an output vector for every input pattern
- Vector indicates normalised distance from stored patterns
HTM Example

The Pictures problem
• Visual Pattern Recognition problem
• Binary image 32x32 pixel
• Create movies of images
• Using all transformations system should be invariant to
  – Translation
  – Rotation
  – Scale variations
Pictures Task

(A)

- dog
- helicopter
- table lamp

(B)

- Image Number
- Euclidean Distance
- dog, helicopter, table lamp

Graz, 27.4.2009
Pictures Network Structure

Level 3

Nodes

Level 2

d

c

Level 1

a
b

Input Image

32 pixels

4x4 pixel patches from input image

32 pixels

CI Seminar A
Example Operation of nodes

- Train first layer
- Switch first layer to sensing/inference mode
- Train second layer
- …
Noise / Ambiguity

• Complicate learning process
• Nodes have to store large number of patterns
• Requires modification
• Layer 1: Noise: Pre-clustering (k-means)
• Higher Layer: Ambiguity: Choose winner
Picture Example Demo
Comparison to SVM

- Classify CAPTCHAs
- Each letter 64x64 pixel
- Four-layer HTM
- 30 examples of each letter
- ~99% accuracy

- 120 pictures
  - Slightly distorted
  - Polynomial distorted
  - Radial distorted
Performance on CAPTCHAs

- SVM 9% - HTM 20%
- Reasonable performance
- Large computational effort (~20h)
Criticism

• Not really a scientific paper (no peer review, no references)
• Absurd claims in presentations
• Inconsistencies between papers and presentations
• HTM is form of Bayesian network
  – Exception: self-training, Parent-child relationship ...
• Nothing new but some modifications
Conclusion

• Copy behavior from biology
  – Generic algorithm/structure
  – Try to deduce functionality
• Combine with existing solutions
• But don't neglect mathematical background
Thank you for your attention!

Any questions?
References

„On Intelligence“, J. Hawkins, S. Blakeslee, 2004


„The HTM learning algorithms“, D. George, B. Jaros, 2007


“Using Numenta’s hierarchical temporal memory to recognize CAPTCHAs“, Y. J. Hall, R. E. Poplin, 2007