Discussion points

Gregor Schöner
Institute for Neural Computation (INI)
dynamicfieldtheory.org

embedding DFT

- DFT, embodiment, dynamical systems thinking
- DFT and connectionism
- DFT and computational neuroscience
- DFT and Deep Neural Networks
- DFT and learning
- DFT and Spiking Neural Networks/ Neuromorphics

contrasting DFT

- DFT vs computational models
- DFT vs cognitive architectures (ACT-R/SOAR etc)
- DFT vs neural cognitive architectures (LIDA, Dora, Leabra, DAC)
- DFT vs VSA

embedding DFT

- DFT, embodiment, dynamical systems thinking
- DFT and connectionism
- DFT and computational neuroscience
- DFT and Deep Neural Networks
- DFT and learning
- DFT and Spiking Neural Networks/ Neuromorphics

Embodiment

- emphasizes the sensori-motor origin of cognition in evolution and development…
- sometimes interpreted to be supported by activation of motor systems during mental operations
- but: that is not mandatory…

Embodiment hypothesis of DFT

- 1) sensory-motor behavior involves a lot of cognition

- attention/gaze
- active perception/working memory
- action plans/decisions/ sequences
- motor control
- background knowledge
- learning from experience

Embodiment hypothesis of DFT

- 2) the dynamic properties of sensory-motor behavior:
  - continuous state, continuous time, stability ..
  - continuous/intermittent link to the sensory and motor surfaces
- are inherited by (higher) cognition

- => cognition is generated in specific embodied cognitive architectures that emerged from evolution/development

Dynamical Systems Thinking (DST)

- Thelen, Smith and many others
- DST is essentially the metaphorical use of dynamical systems ideas that DFT formalizes

embedding DFT

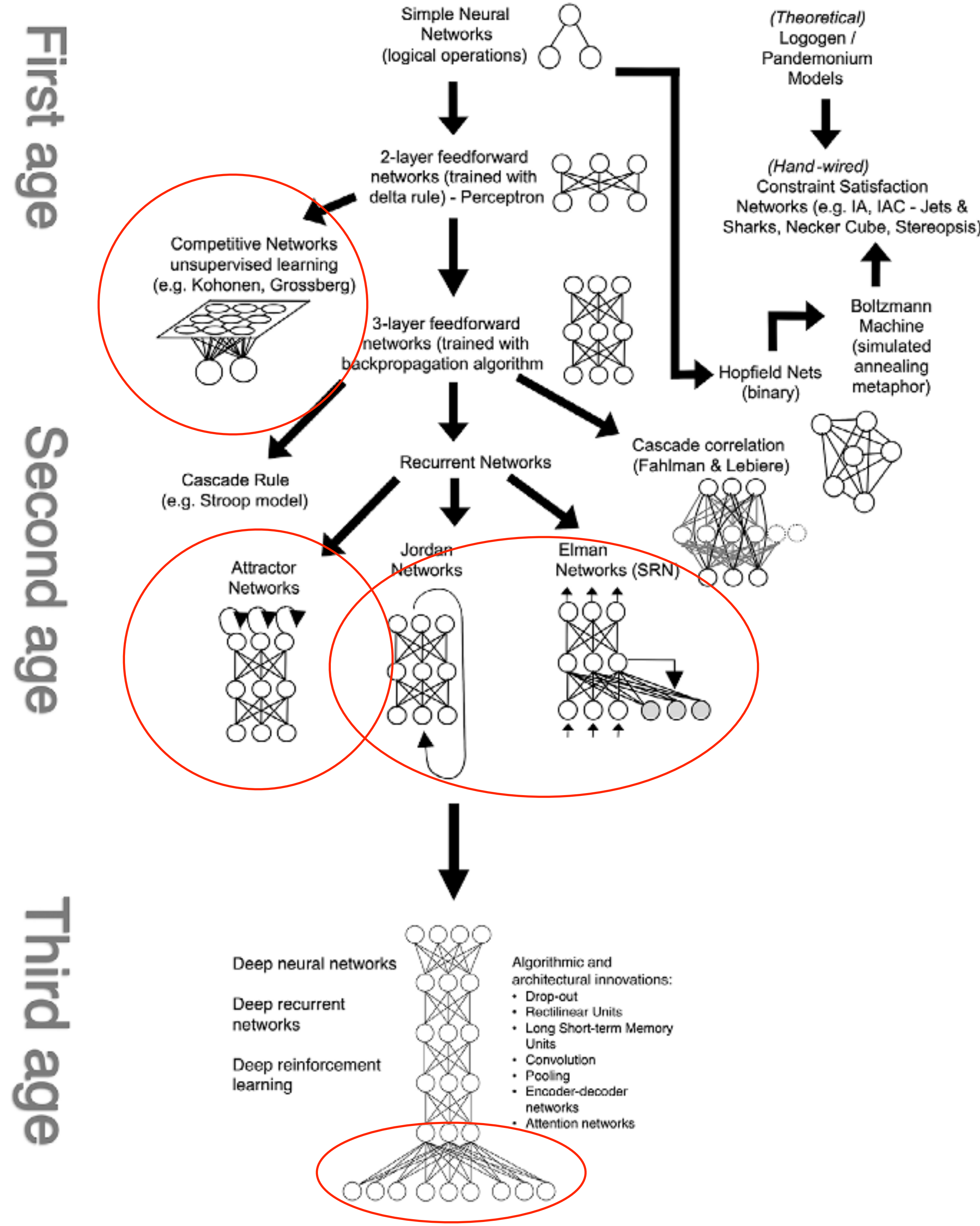
- DFT, embodiment, dynamical systems thinking
- DFT and connectionism
- DFT and computational neuroscience
- DFT and Deep Neural Networks
- DFT and learning
- DFT and Spiking Neural Networks/ Neuromorphics

DFT and connectionism

- DFT models are neural network models in the most general sense…
- and share with these the level of description
  - continuous activation
  - sigmoid threshold function (replacing spiking)

Connectionism

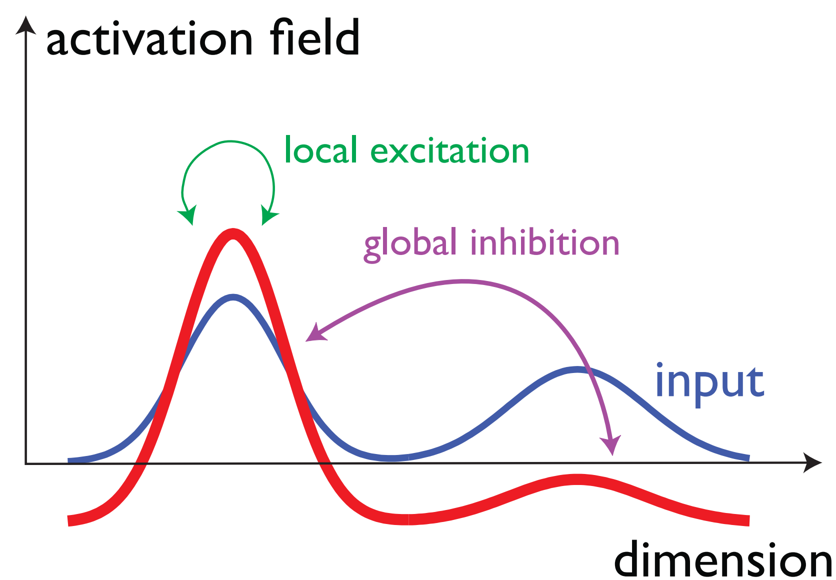
- DFT models are recurrent neural networks
- in continuous time and continuous space limit



[Thomas, McClelland, 2023]

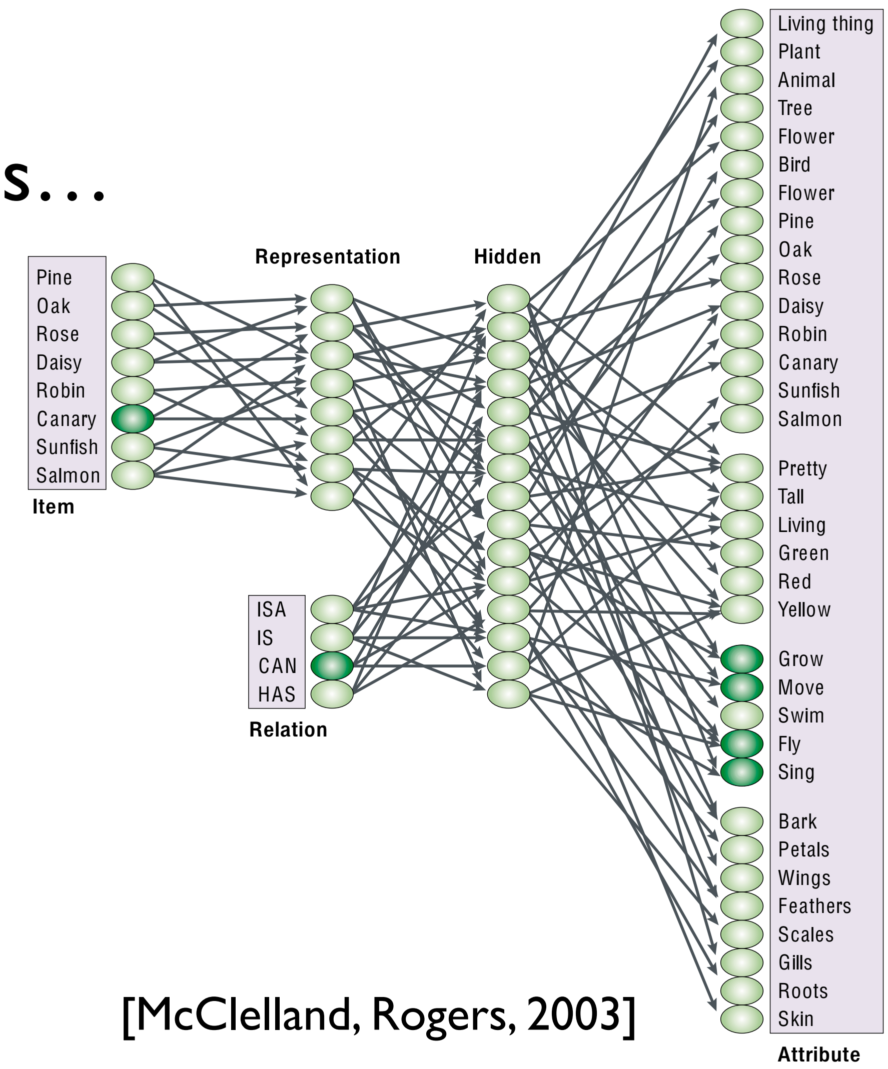
DFT makes more specific commitments than connectionism

- stability of functionally significant states
- instabilities as key elements of neural processing .. sequences
- => all autonomous cognition is based on localist representations



DFT makes more specific commitments than connectionism

- scaling argument => all cognitive representations are low-dimensional
- no distributed representations…
- no association !
- binding across localist representations replaces association in DFT



[McClelland, Rogers, 2003]

road map

embedding DFT

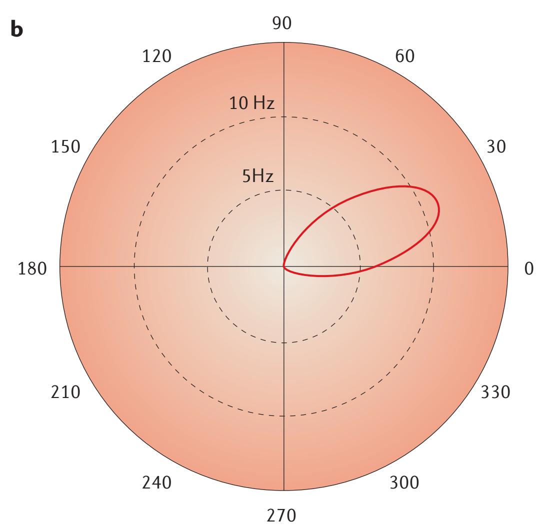
- DFT, embodiment, dynamical systems thinking
- DFT and connectionism
- DFT and computational neuroscience
- DFT and Deep Neural Networks
- DFT and learning
- DFT and Spiking Neural Networks/ Neuromorphics

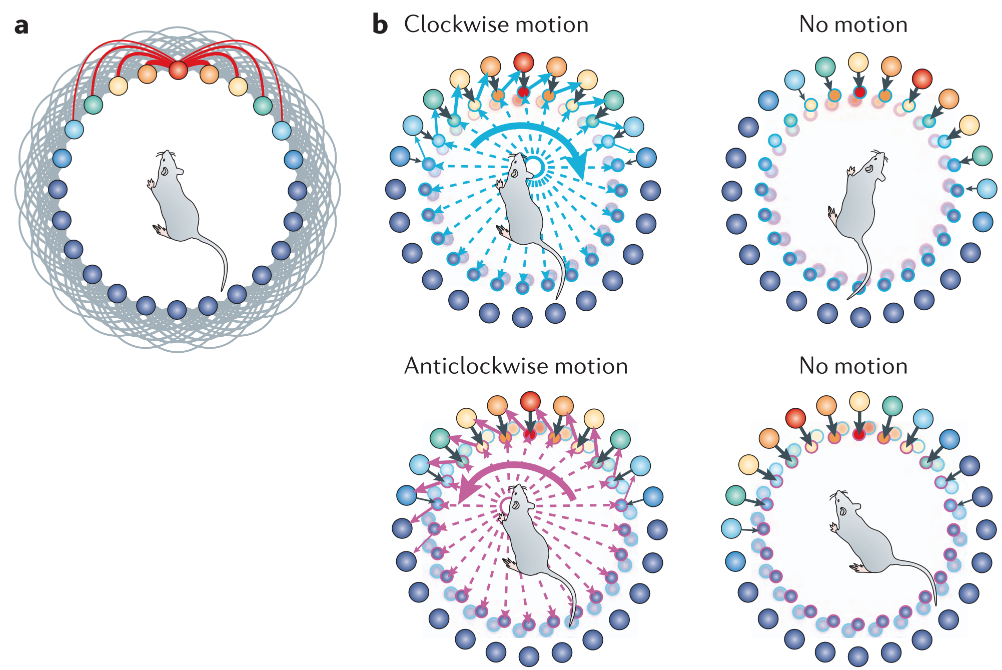
DFT and computational neuroscience

- computational neuroscience takes the neural mechanistic foundations more seriously than connectionism (and than DFT)
- was the setting from which original ideas of attractor dynamics in neural networks arose: Wilson, Cowan, 1972, 73; Amari 1977
- much current work that aims to understand mechanistic basis for neural function
- typically seeking neural evidence as a constraint

Example: Neural attractor dynamics for head orientation

- Neural evidence for head-orientation cells…
- Neural attractor dynamics (neural field) for heading direction: estimate/working memory of current orientation, updated by integrating motor commands…

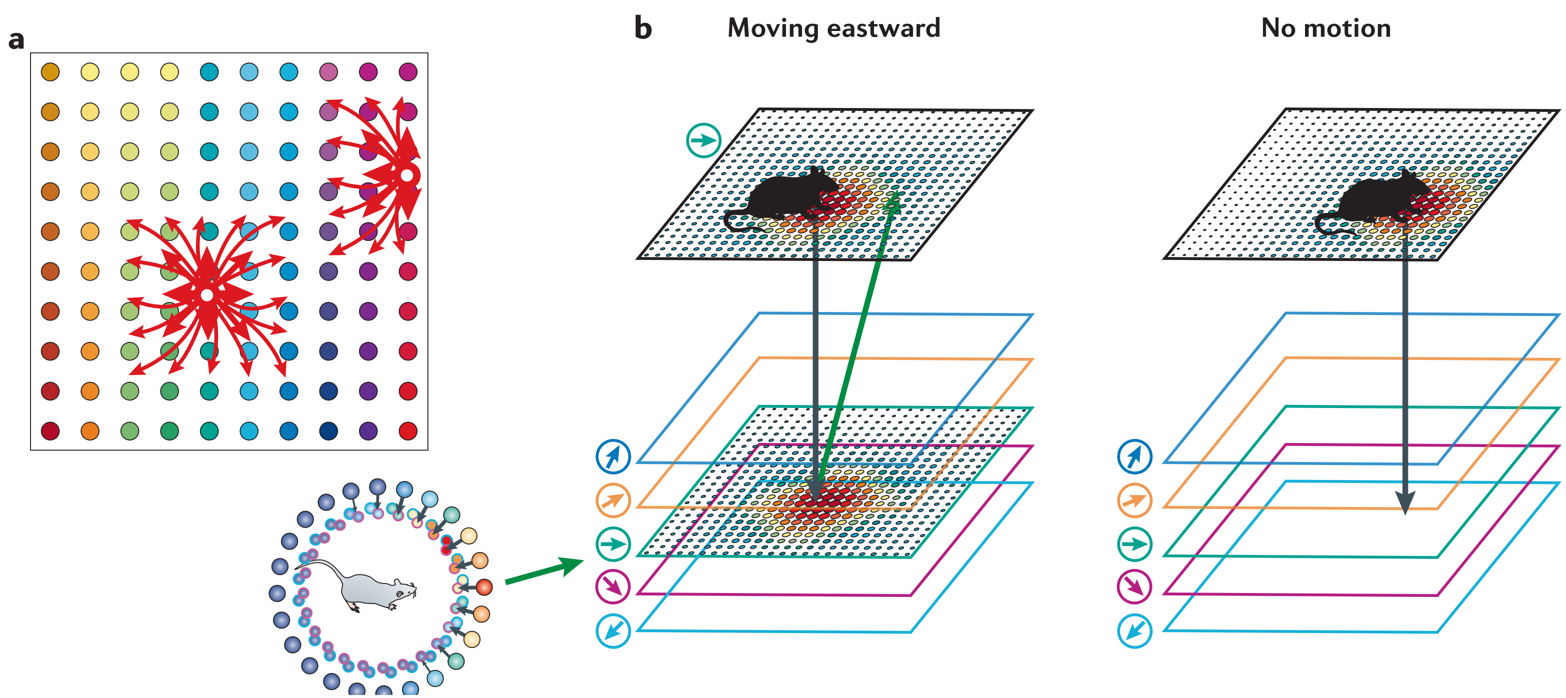




[McNaughton et al., *Nature reviews neuroscience* 2006]

Example: Neural attractor dynamics for head orientation

- Extension to spatial map of ego-position using slice input and directed connectivity
- to account for place cells in HC



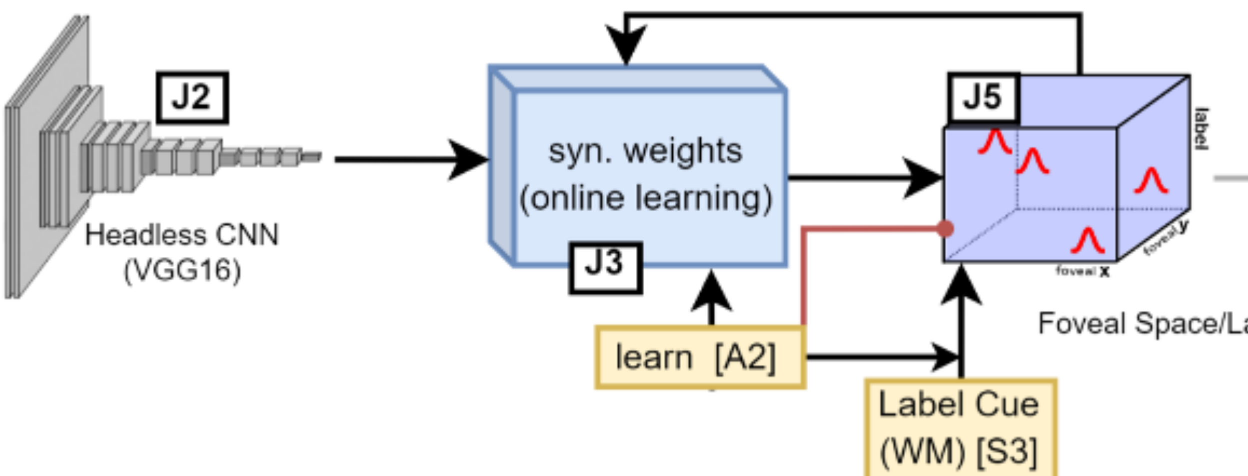
[McNaughton et al., *Nature reviews neuroscience* 2006]

embedding DFT

- DFT, embodiment, dynamical systems thinking
- DFT and connectionism
- DFT and computational neuroscience
- DFT and Deep Neural Networks
- DFT and learning
- DFT and Spiking Neural Networks/ Neuromorphics

DFT and DNN

- DNN: the apparent high-dimensionality/distributed representation gives discriminatory power to DNN
- but only effective in the presence of input riven by sensory inputs => no actual cognition!
- all cognition takes place in the "read-out" layer = competing neural nodes
- => Raul Grieben's lecture



[Raul Grieben]

embedding DFT

- DFT, embodiment, dynamical systems thinking
- DFT and connectionism
- DFT and computational neuroscience
- DFT and Deep Neural Networks
- DFT and learning
- DFT and Spiking Neural Networks/ Neuromorphics

DFT and DNN

- most "learning" in Neural Network modeling (including Deep Learning) is actually "fitting"
  - obvious for supervised learning
  - even unsupervised learning provides examples from the outside
- autonomous learning: learning from experience
  - is only accessible if there is autonomous behavior that generates the experience..
  - and that is what DFT enables…
- a research challenge
  - first inroads by Sandamirskaya (2014), Tekülve, Schöner (2020)

road map

embedding DFT

- DFT, embodiment, dynamical systems thinking
- DFT and connectionism
- DFT and computational neuroscience
- DFT and Deep Neural Networks
- DFT and learning
- DFT and Spiking Neural Networks/ Neuromorphics

DFT and SNN/neuromorphics

- DFT~mean field theory in the population picture
  - activation is something like a population level membrane potential
- Spiking brings in new properties not captured in this approximation
  - spike timing: spikes as synchronicity detector
  - sparseness: low correlations between neurons
- spiking as a form of "implementation"
- Neuromorphics makes use of the spiking concept

roadmap

contrasting DFT

- DFT vs computational models
- DFT vs cognitive architectures (ACT-R/SOAR etc)
- DFT vs neural cognitive architectures (LIDA, Dora, Leabra, DAC)
- DFT vs VSA

DFT and computational models/theory

- David Marr’s levels
  - computational
  - algorithmic
  - implementation

Computational level

- “computation”in the sense: given input, determine the output… => “computational laws” of vision, action, cognition…
- probabilistic approaches such as Bayes networks reside at the computational level
- normative models such as optimal estimation, optimal control..
- … currently influential ideas…

Computational level

- example: given the optic flow from a rigid environment through which the observer moves, the observer's ego-motion can be computed (up to a scaling factor)

[Robert, Potthast, Dellaert, 2009]

Computational level

- “describe” neural function rather than “explain” the underlying process

Algorithmic level

- example: estimate the optic flow by searching through two subsequent images and finding corresponding pairs of locations
- information processing model of cognition… are algorithmic accounts
- "pseudo-code" descriptions of computational models are algorithmic accounts

Implementation level

- in human cognition: neural process models
  - potentially at different levels… from abstract connectionist to computational neuroscience models
- in AI: the concrete numerical implementations of algorithms, or probabilistic models etc.

roadmap

contrasting DFT

- DFT vs computational models
- DFT vs cognitive architectures (ACT-R/SOAR etc)
- DFT vs neural cognitive architectures (LIDA, Dora, Leabra, DAC)
- DFT vs VSA

DFT and cognitive architectures: ACT-R, SOAR etc

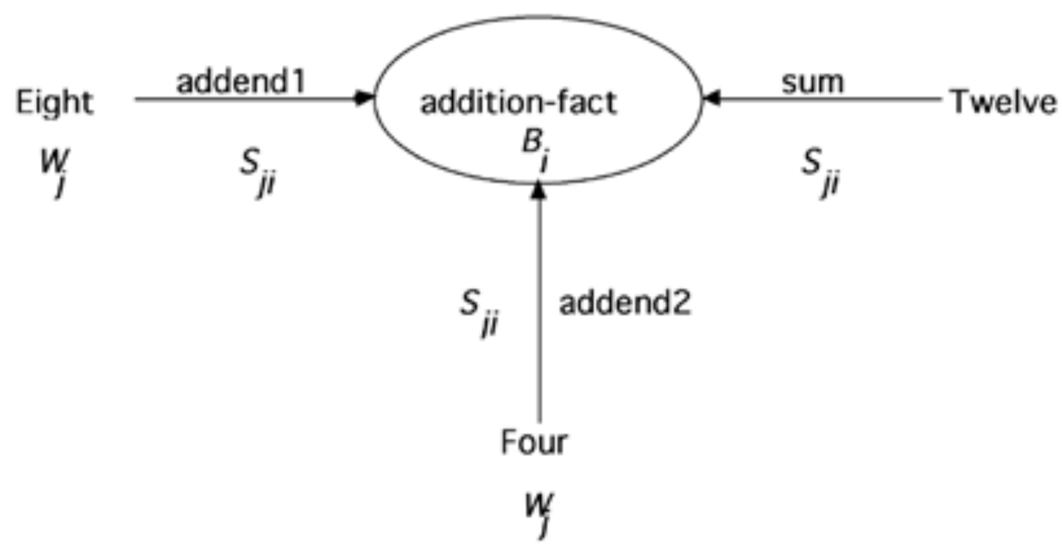
- aligned with early AI … Herbert Simon… Allen Newell: "general intelligence"
- the "computer metaphor": cognition consists of the manipulation of symbols… constrained by rules/programs

Modularity

- computational elements are defined by their input/output interface
- they are "impenetrable" so that their inner states do not affect other modules… Fodor, Pylyshyn
  - related to the AI notion of "encapsulation"
- => understanding cognition = understanding how link among modules through their input/output interfaces
- => cognitive architectures

ACT-R elements: chunks

- represent "facts": memory items, perceived items, motor commands, rules, operations (contents)
- graded, time-varying activations
  - control if chunk is instantiated
  - determines which other chunks are instantiated

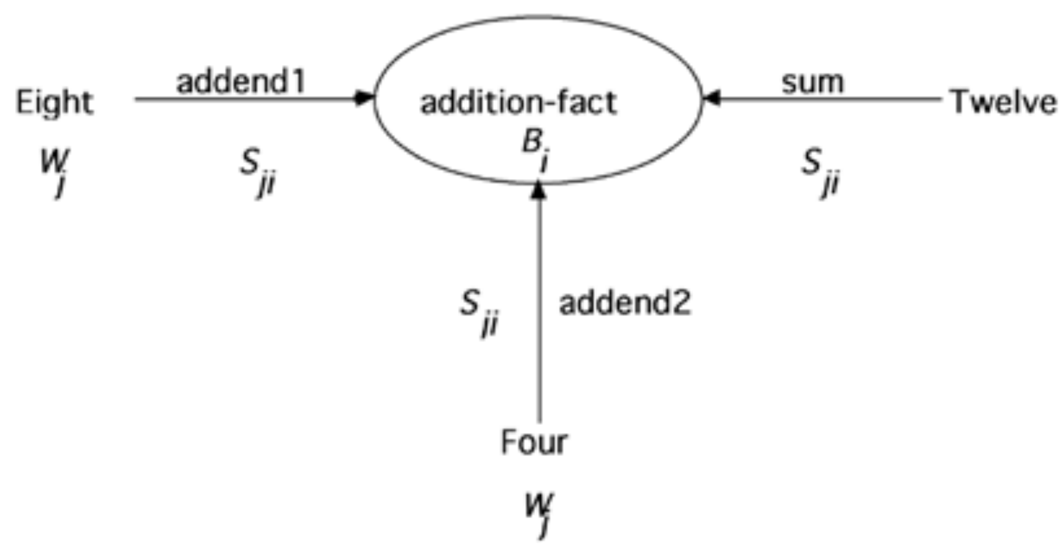


[Anderson, 2007]

. A representation of a chunk with its subsymbolic quantities.

ACT-R chunks

- chunk activation above threshold => a production "fires"
- chunk content is "executed"
  - e.g. an addition is performed



[Anderson, 2007]

. A representation of a chunk with its subsymbolic quantities.

Modular architectures

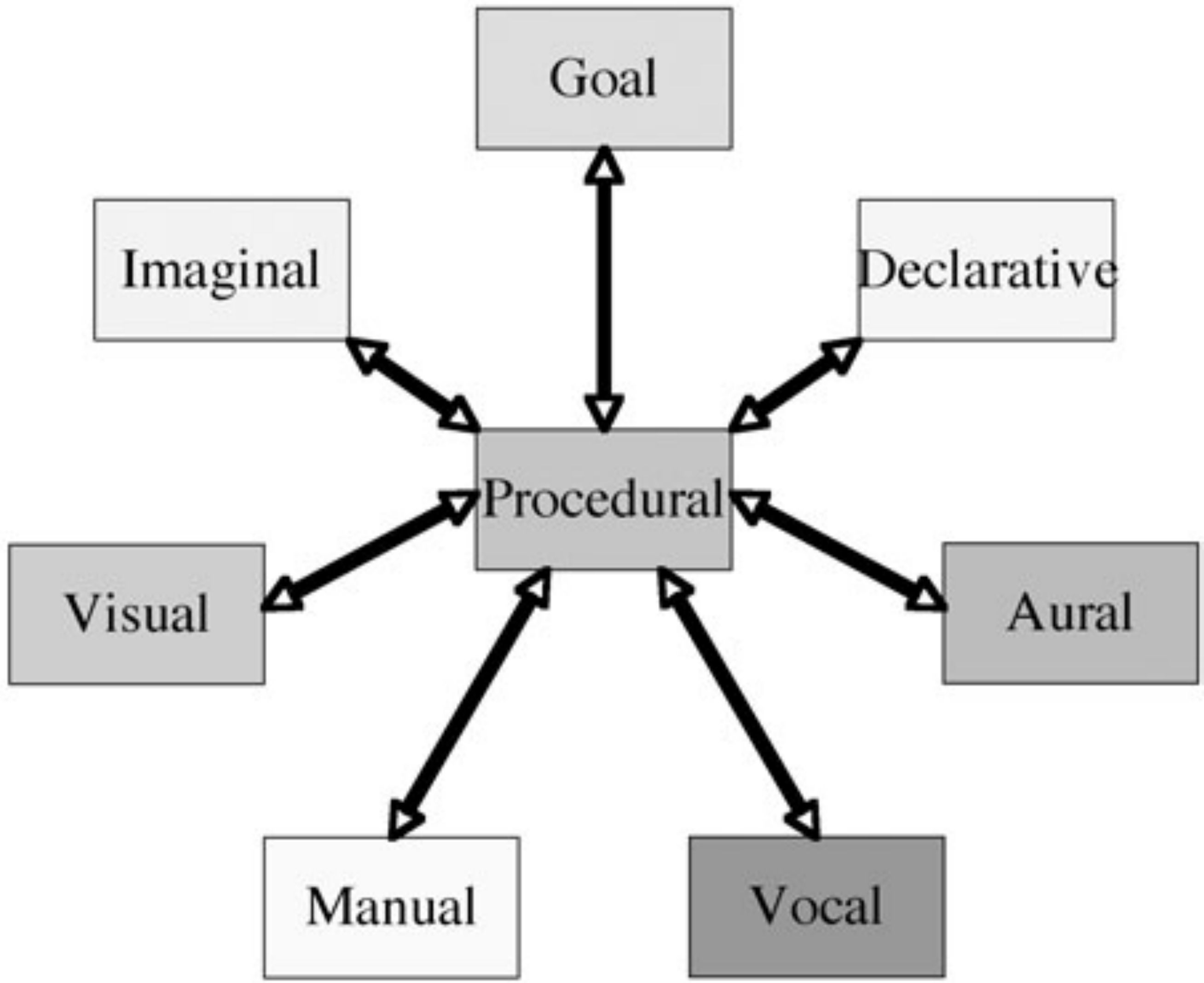
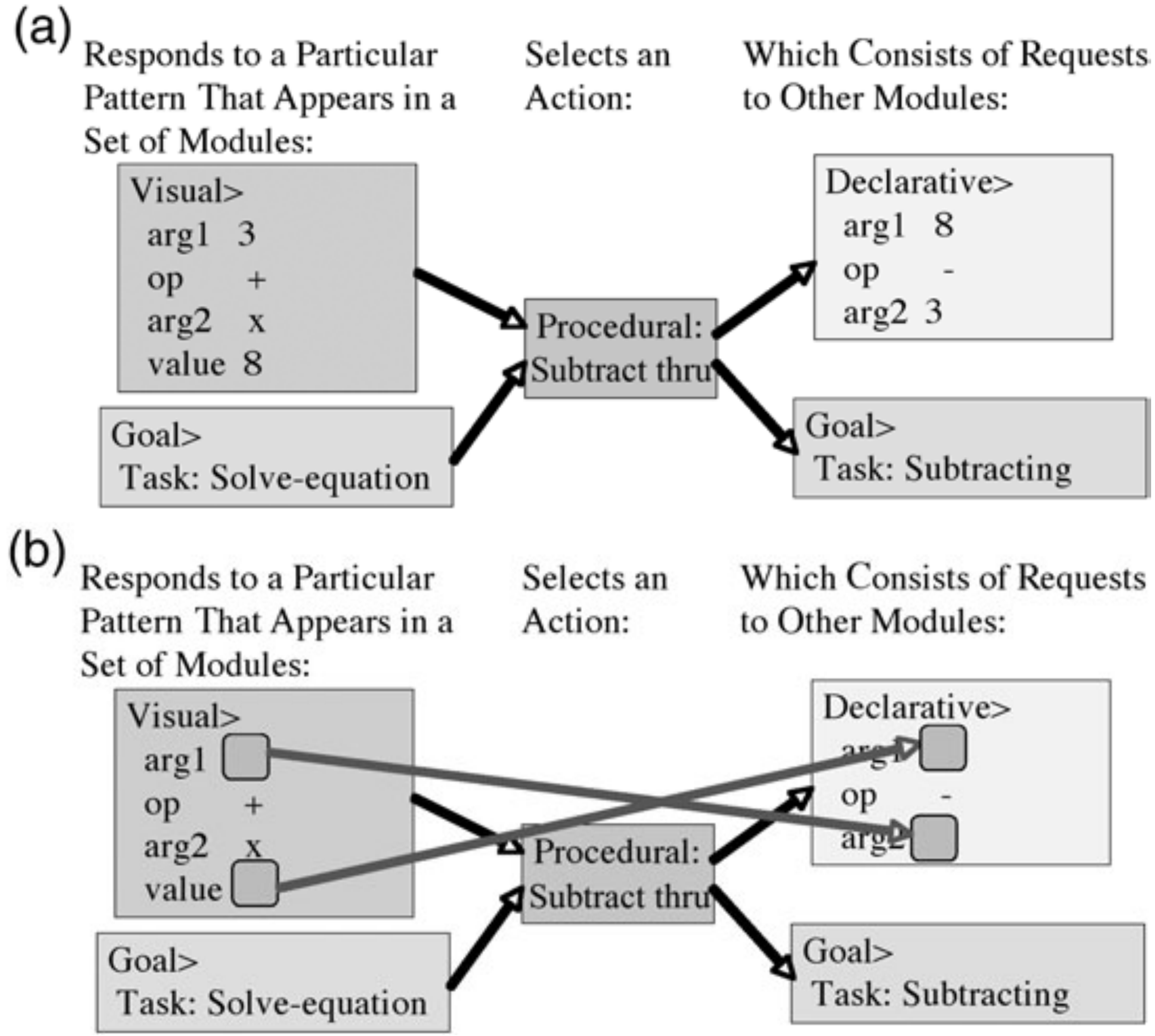


Figure 2.2. The modules implemented in ACT-R 6.0.

Production rule: "computation"



Evidence by comparing "computational effort" of model to human experiment

- exemplary problem: mental arithmetic

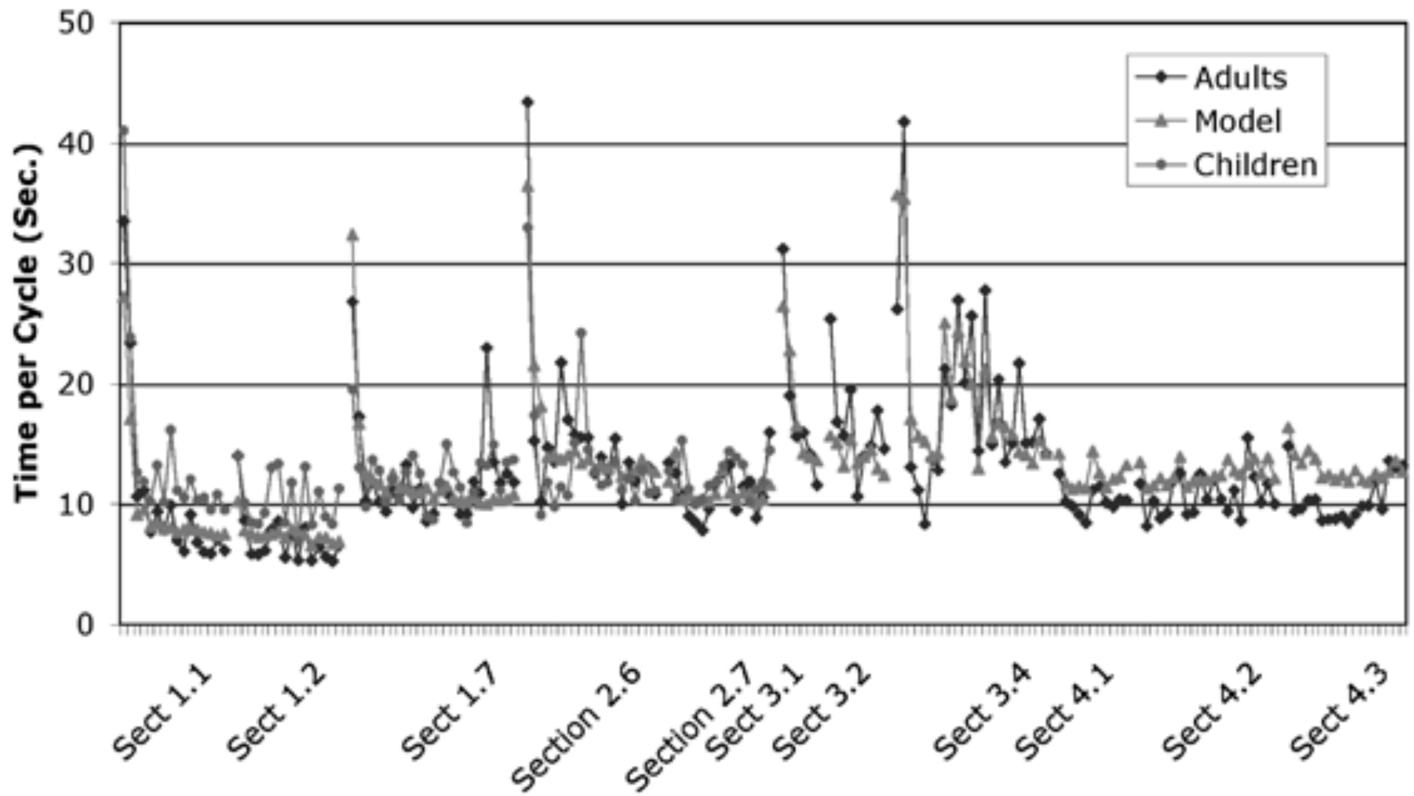


Figure 5.7. A comparison of the performance of the model with that of children learning the linear form of algebra and adults learning the data-flow form.

DFT and cognitive architectures

- DFT: connectivity specifies both “content” and instantiation of representation
- DFT: autonomous evolution of activation… leads to events through instabilities … vs. is controlled by computational cycle
- DFT: constraints emerge from nature of neural dynamics… vs. is imposed to fit data

roadmap

contrasting DFT

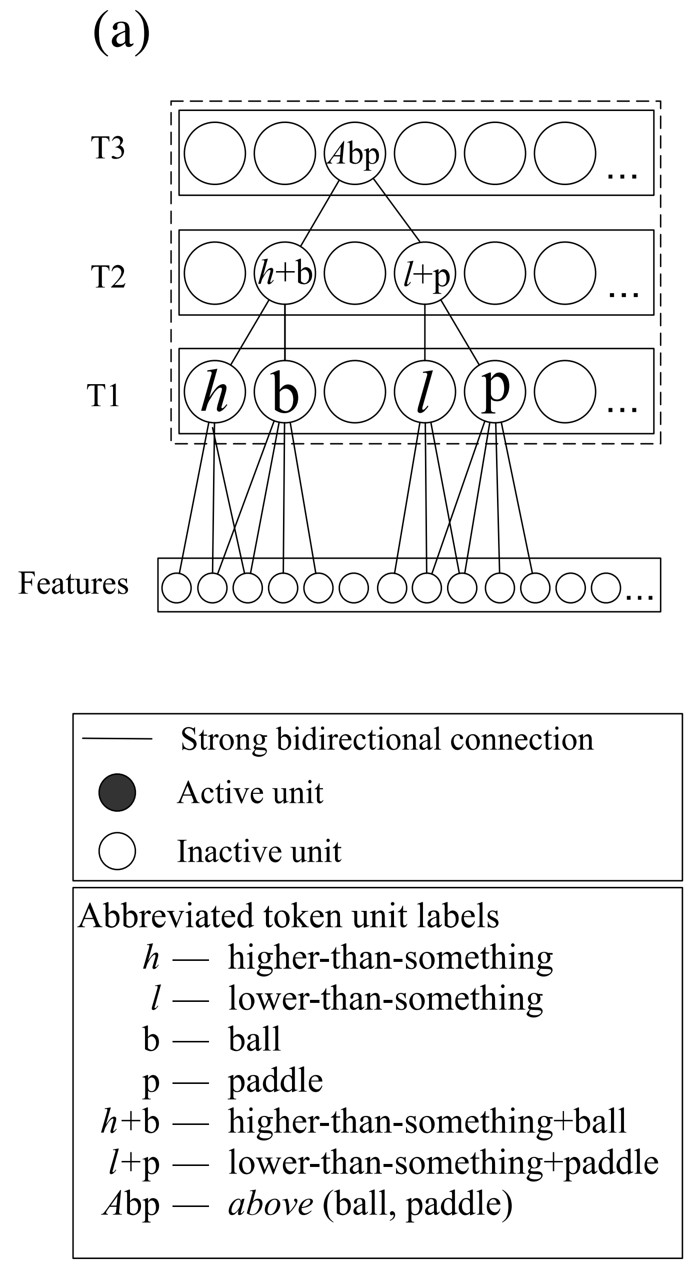
- DFT vs computational models
- DFT vs cognitive architectures (ACT-R/ SOAR etc)
- DFT vs neural cognitive architectures (LIDA, Dora, Leabra, DAC)
- DFT vs VSA

DFT vs neural cognitive architectures (Lida, Dora, Leabra, DAC …)

- share principles of neural representation
- in many cases, the processing itself is algorithmic (DAC and others)
- in other cases, the actual cognitive operations are information processing (Leabra)

Dora

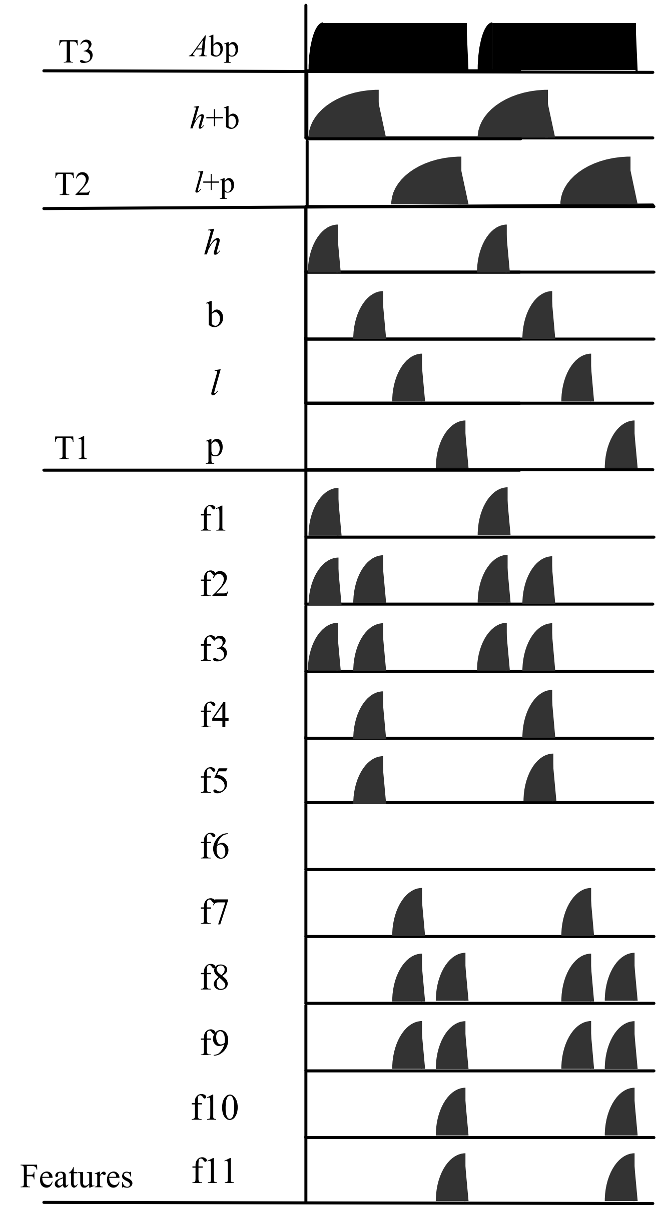
- interesting approach that maximally overlaps with DFT
- different principle of binding…
- that scales poorly with number of concepts



[Doumas et al., Psych Rev 2020]

Dora

- autonomous processing: period and hierarchically nested timing
- ~neural dynamics
- but lacks stability and invariance when elementary processing steps take different amounts of time



[Doumas et al., Psych Rev 2020]

roadmap

contrasting DFT

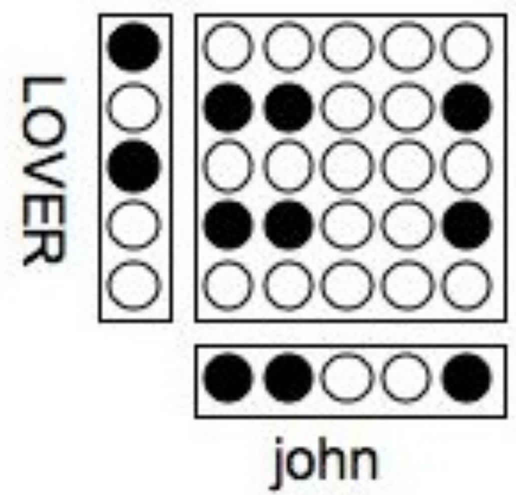
- DFT vs computational models
- DFT vs cognitive architectures (ACT-R/SOAR etc)
- DFT vs neural cognitive architectures (LIDA, Dora, Leabra, DAC)
- DFT vs VSA

DFT vs VSA

- Vector-symbolic architectures (VSA): an (alternative) neural account for higher cognition
- in the original version (Smolensky): role-filler binding… compatible with DFT

VSA

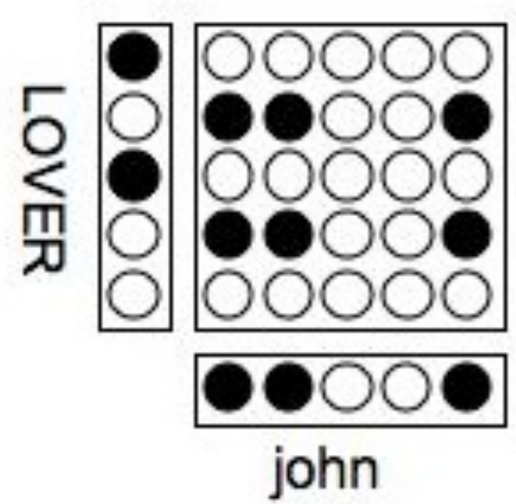
- each concept is represented by an activation vector
  - column vectors $x_{\text{John}}, x_{\text{Mary}}, \cdots$
  - column vectors, $y_{\text{LOVER}}, y_{\text{BELOVED}}$
- requires that the symbol grounding problem is solved at encoding/decoding



[Levy, Gayler, 2008]

binding in VSA

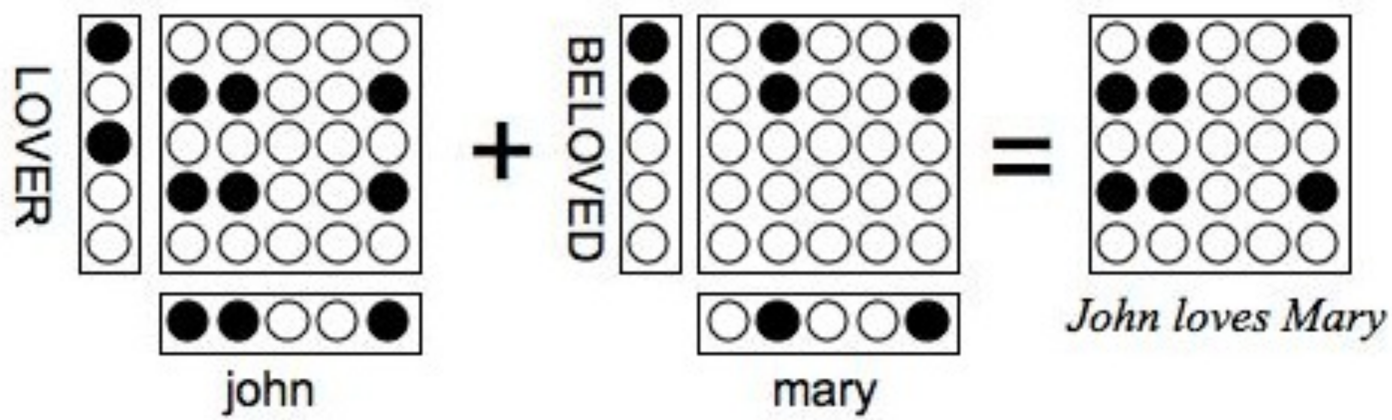
- binding: make an array through direct product
- $x_{\text{John}} \cdot y_{\text{LOVER}}^T$
  - this increases the dimension



[Levy, Gayler, 2008]

bundling in VSA

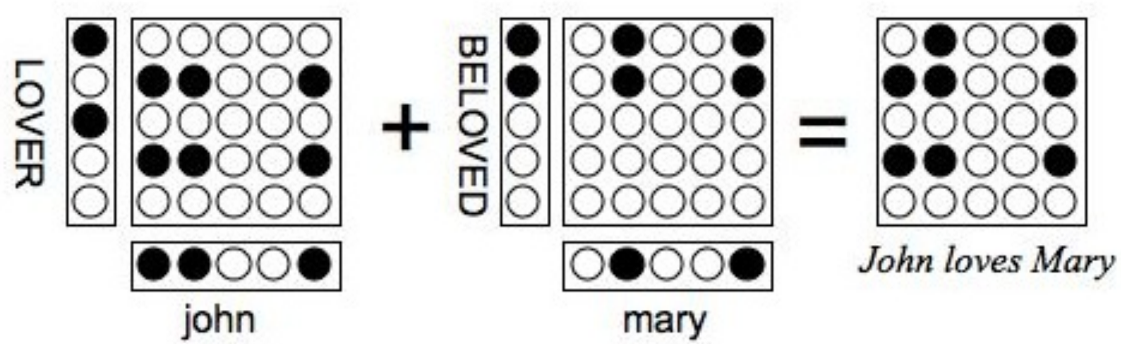
- done simply by adding the matrices..



[Levy, Gayler, 2008]

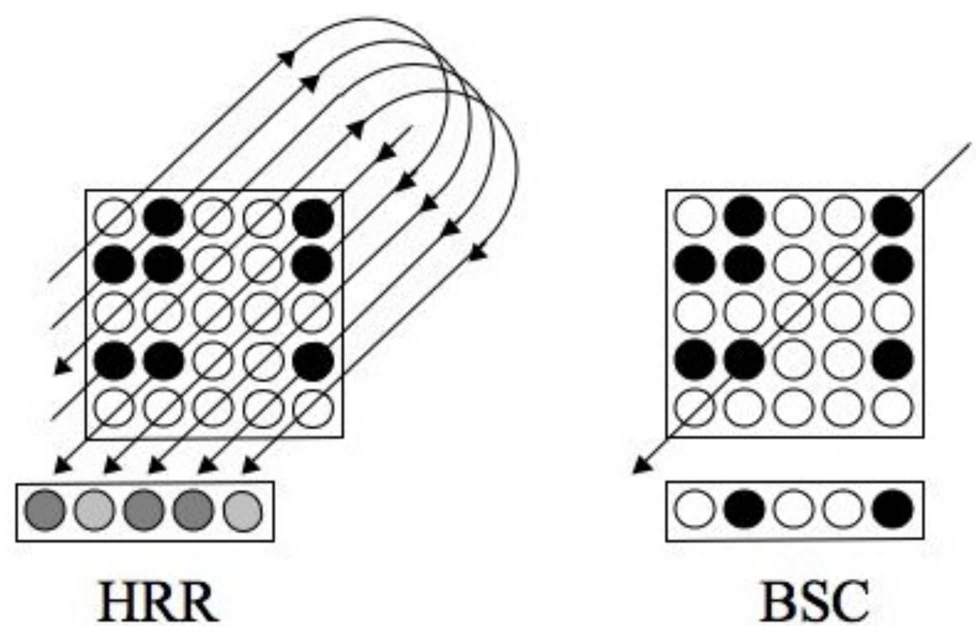
dimensionality reduction

- to enable continued, even recursive application of the binding operation, the growth in dimension has to be stopped
- this works if there is “redundancy” the bound representation.. which is true for random vectors/distributed representations



dimensionality reduction

- the *holographic* method (due to Plate, HRR): sum along diagonals… a convolution…
- the *block splatter* (BSC) method: just take the diagonal



[Levy, Gayler, 2008]

unbinding

- a form of inverse
- $x \otimes y = x \cdot y^T$
- => $(x \otimes y) \cdot y = x \cdot (y^T \cdot y) = x \|y\|^2$
- similarly
  $x^T \cdot (x \otimes y) = (x^T \cdot x) \cdot y^T = y^T \|x\|^2$
- so recover original vector up to a norm

clean-up

- due to compression, the inverse is not exact
- need to clean-up=restore the original vector…
- by auto-association
  - e,.g. the vectors as attractors of a Hopfield network, so that you only need to get into the basin of attraction,…

How does VSA operate?

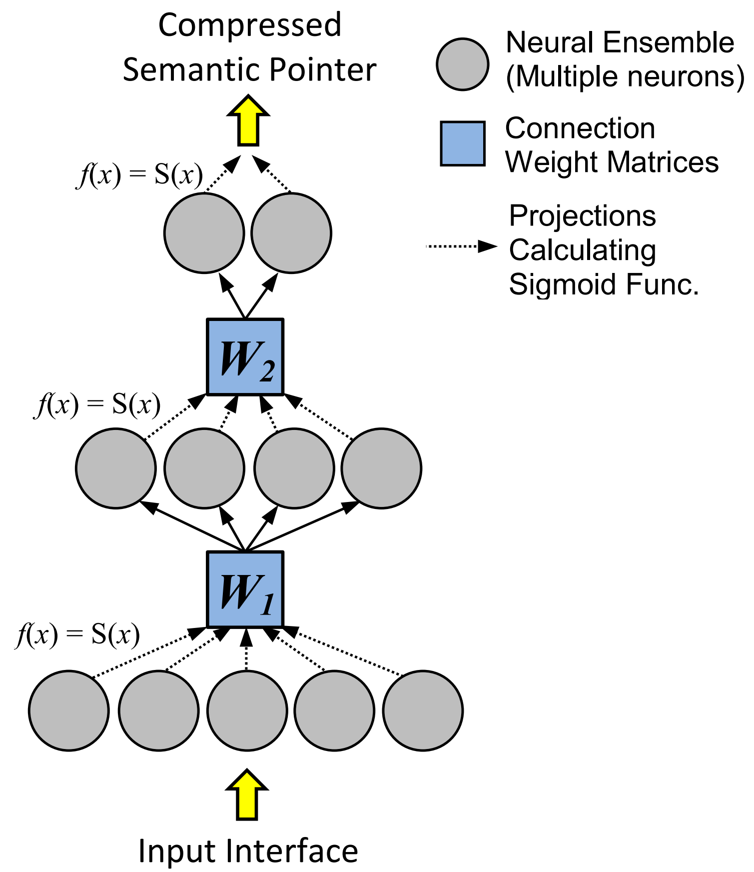
- encode fillers/roles as vectors = symbols, and provide them as input…
  - VSA (vector-symbolic architecture) then binds/bundles/unbinds these sequentially as defined by the VSA
  - output symbols.. that can decoded
  - encoding/decoding not part of VSA (but can be done with NN)
- autonomous organization of sequence not part of VSA
- stabilizing the high-dimensional vectors is not trivial

DFT vs NEF

- Eliasmith's Neural Engineering Framework (NEF) as a possible neural implementation of VSA
  - vectors represented by (small) populations of spiking neural networks
- NEF is "model neutral"… essentially a method to "numerically" implement any neural model

DFT vs VSA

- But: to preserve the original vectors, connectivity in VSA/NEF (SPAUN) architectures is very special: decode and re-encode..
- => SPAUN brains are not robust against learning/development due to non-local inter-dependence of connectivities
- (and other issues)



[Choo Feng Xuan, 2018]

roadmap

contrasting DFT

- DFT vs computational models
- DFT vs cognitive architectures (ACT-R/SOAR etc)
- DFT vs neural cognitive architectures (LIDA, Dora, Leabra, DAC)
- DFT vs VSA

embedding DFT

- DFT, embodiment, dynamical systems thinking
- DFT and connectionism
- DFT and computational neuroscience
- DFT and Deep Neural Networks
- DFT and learning
- DFT and Spiking Neural Networks/ Neuromorphics