APPLIED ARTIFICIAL INTELLIGENCE INSTITUTE
DEAKIN UNIVERSITY

Tutorial at IJCAI, August 19th /20th 2021

Neural Machine Reasoning

Lecture 1: Concepts in neural machine reasoning

Truyen Tran, Vuong Le, Hung Le and Thao Le
{truyen.tran,vuong.le,thai.le,thao.le}@deakin.edu.au

https://neuralreasoning.github.io

Logistics

Truyen Tran

Vuong Le

Hung Le

Thao Le

https://neuralreasoning.github.io

Agenda

- **Introduction**
- Part A: Theory
  - Lecture 1: Concepts in neural machine reasoning
  - Lecture 2: Dual system of reasoning
  - Lecture 3: Neural memories
  - Lecture 4: Reasoning over unstructured sets
  - Lecture 5: Reasoning over graphs
  - Lecture 6: Hybrid neuro-symbolic reasoning

Agenda (cont.)

- Part B: Applications
  - Lecture 7: Neural machine reading comprehension (Text-based QA)
  - Lecture 8:Visual question answering
  - Lecture 9: Video question answering
  - Lecture 10: Combinatorics reasoning

Why still DL in 2021?

Theoretical

- **Expressiveness**: Neural nets can approximate any function.
- **Learnability**: Neural nets are trained easily.
- **Generalisability**: Neural nets generalize surprisingly well to unseen data.

Practical

- **Generality:** Applicable to many domains.
- **Competitive:** DL is hard to beat as long as there are data to train.
- **Scalability:** DL is better with more data, and it is very scalable.

The next AI/ML challenge

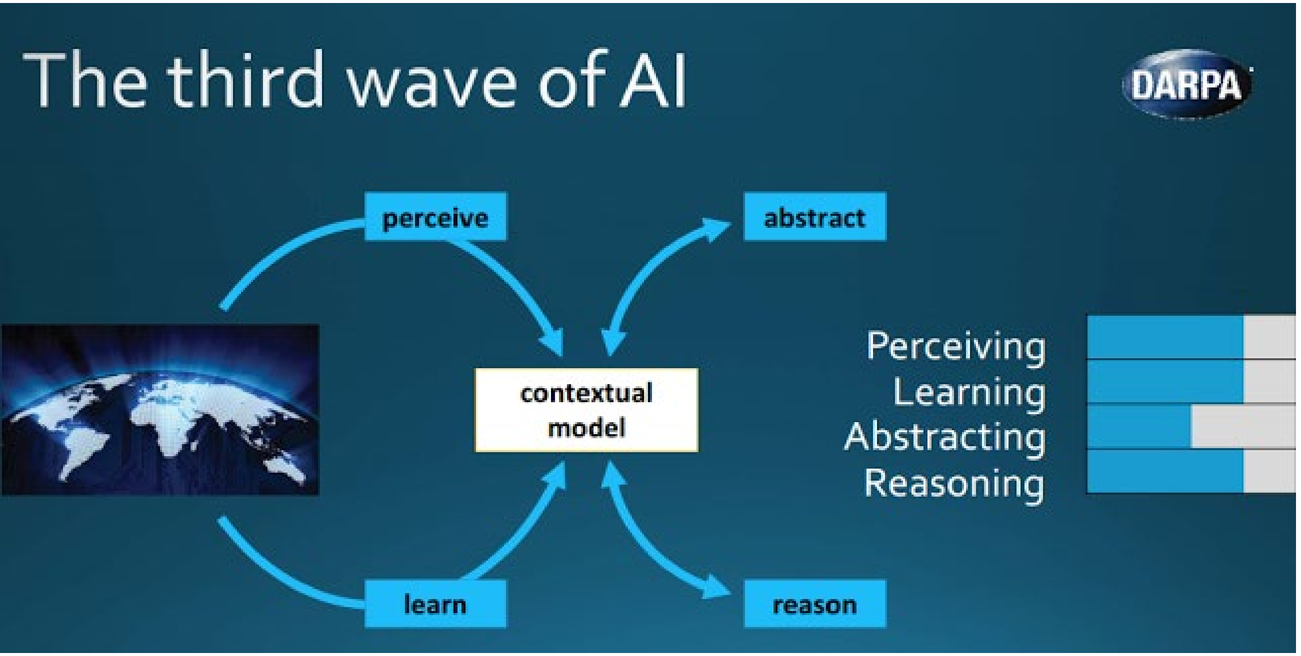


Photo credit: DARPA

2020s-2030s

- Learning + reasoning, general purpose, human-like
- Has contextual and common-sense reasoning
- Requires less data
- Adapt to change
- Explainable

Reasoning in Probabilistic Graphical Models (PGM)

- Assuming models are fully specified (e.g., by hand or learnt)
  - →Estimate MAP as energy minimization
  - →Compute marginal probability
  - →Compute expectation & normalisation constant
- Key algorithm: Pearl's Belief Propagation, a.k.a Sum-Product algorithm in factor graphs.
  - Known result in 2001-2003: BP minimises Bethe free-energy minimization.

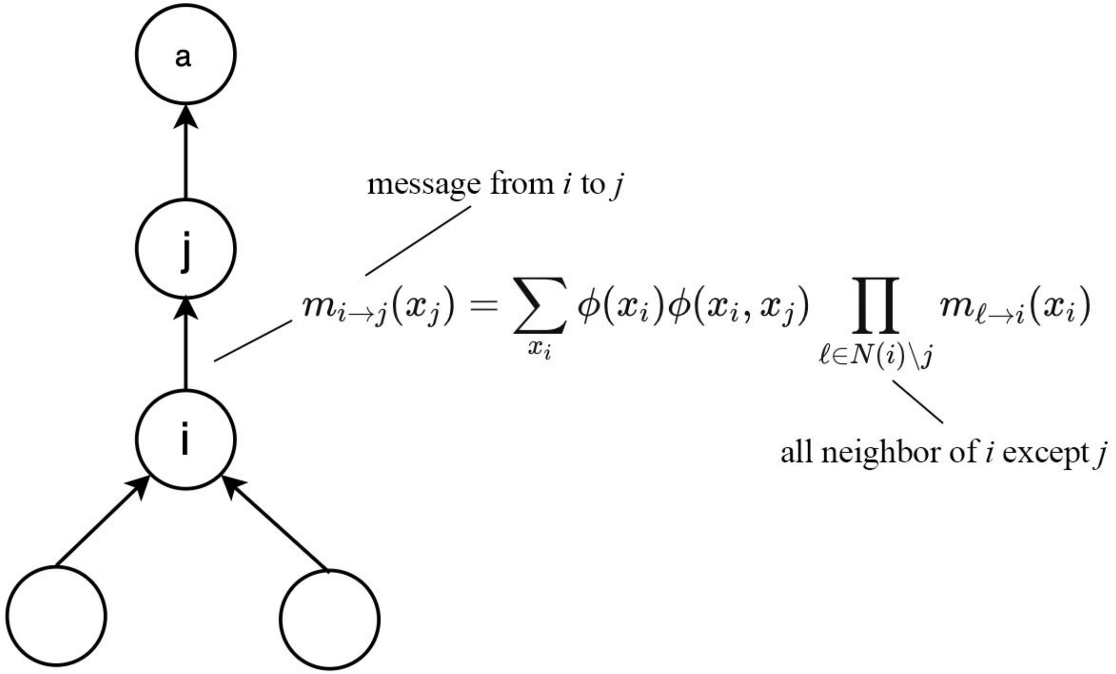


Figure credit: Jonathan Hui

Heskes, Tom. "Stable fixed points of loopy belief propagation are local minima of the bethe free energy." *Advances in neural information processing systems*. 2003.

Agenda

- Introduction
- Part A: Theory
  - **Lecture 1: Concepts in neural machine reasoning**
  - Lecture 2: Dual system of reasoning
  - Lecture 3: Neural memories
  - Lecture 4: Reasoning over unstructured sets
  - Lecture 5: Reasoning over graphs
  - Lecture 6: Hybrid neuro-symbolic reasoning

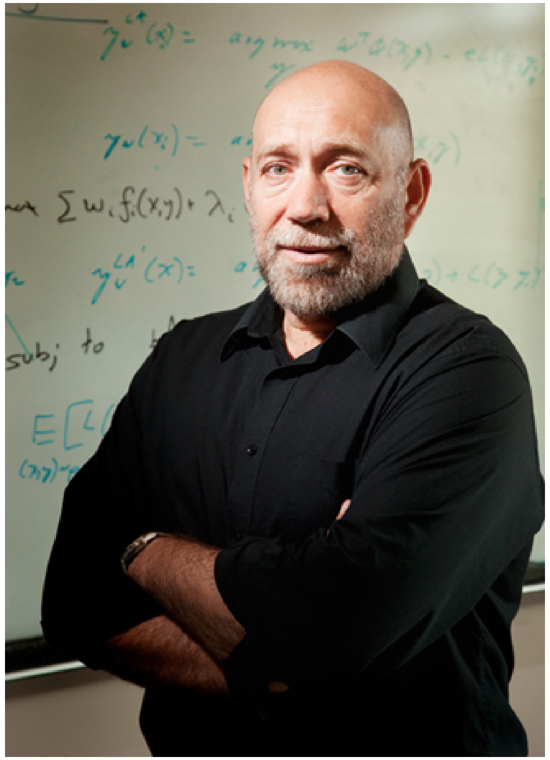
Can we learn to infer directly from data without full specification of models?

Lecture 1: Sub-topics

- Reasoning as a prediction skill that can be learnt from data.
  - Question answering as zero-shot learning.
- Neural network operations for learning to reason:
  - Attention & transformers.
  - Dynamic neural networks, conditional computation & differentiable programming.
- Reasoning as iterative representation refinement & query-driven program synthesis and execution
  - Compositional attention networks.
  - Neural module networks.

Learning to reason

- Learning is to self-improve by experiencing ~ acquiring knowledge & skills
- Reasoning is to deduce knowledge from previously acquired knowledge in response to a query (or a cues)
- Learning to reason is to improve the ability to decide if a knowledge base entails a predicate.
  - E.g., given a video f, determines if the person with the hat turns before singing.



(Dan Roth; *ACM Fellow; IJCAI John McCarthy Award*)

Khardon, Roni, and Dan Roth. "Learning to reason." *Journal of the ACM (JACM)* 44.5 (1997): 697-725.

Learning to reason, a definition

Definition 2.1.1. An algorithm A is an *exact reasoning* algorithm for the reasoning problem $(\mathcal{F}, \mathcal{Q})$, if for all $f \in \mathcal{F}$ and for all $\alpha \in \mathcal{Q}$, when A is presented with input (f, α), A runs in time polynomial in n and the size of f and α, and answers "yes" if and only if $f \models \alpha$.

E.g., given a video f, determines if the person with the hat turns before singing.

Khardon, Roni, and Dan Roth. "Learning to reason." *Journal of the ACM (JACM)* 44.5 (1997): 697-725.

Practical setting: `(query,database,answer)` triplets

- This is very general:
  - Classification: Query = *what is this?* Database = *data.*
  - Regression: Query = *how much?* Database = *data.*
  - QA: Query = *NLP question.* Database = *context/image/text.*
  - Multi-task learning: Query = *task ID.* Database = *data.*
  - Zero-shot learning: Query = *task description.* Database = *data.*
  - Drug-protein binding: Query = *drug*. Database = *protein.*
  - Recommender system: Query = *User (or item)*. Database = *inventories (or user base)*;

Can neural networks reason?

Reasoning is not necessarily achieved by making logical inferences

There is a continuity between [algebraically rich inference] and [connecting together trainable learning systems]

Central to reasoning is composition rules to guide the combinations of modules to address new tasks

"When we observe a visual scene, when we hear a complex sentence, we are able to explain in formal terms the relation of the objects in the scene, or the precise meaning of the sentence components. However, there is no evidence that such a formal analysis necessarily takes place: we see a scene, we hear a sentence, and we just know what they mean. This suggests the existence of a middle layer, already a form of reasoning, but not yet formal or logical."

Bottou, Léon. "From machine learning to machine reasoning." *Machine learning* 94.2 (2014): 133-149.

Hypotheses

- Reasoning as just-in-time **program synthesis**.
- It employs **conditional computation**.
- Reasoning is **recursive**, e.g., mental travel.

Two approaches to neural reasoning

- **Implicit chaining of predicates through recurrence:**
  - Step-wise query-specific attention to relevant concepts & relations.
  - Iterative concept refinement & combination, e.g., through a working memory.
  - Answer is computed from the last memory state & question embedding.
- **Explicit program synthesis:**
  - There is a set of modules, each performs an pre-defined operation.
  - Question is parse into a symbolic program.
  - The program is implemented as a computational graph constructed by chaining separate modules.
  - The program is executed to compute an answer.

In search for basic neural operators for reasoning

- Basics:
  - Neuron as feature detector → Sensor, filter
  - Computational graph → Circuit
  - Skip-connection → Short circuit
- Essentials
  - Multiplicative gates → AND gate, Transistor, Resistor
  - Attention mechanism → SWITCH gate
  - Memory + forgetting → Capacitor + leakage
  - Compositionality → Modular design
- ..

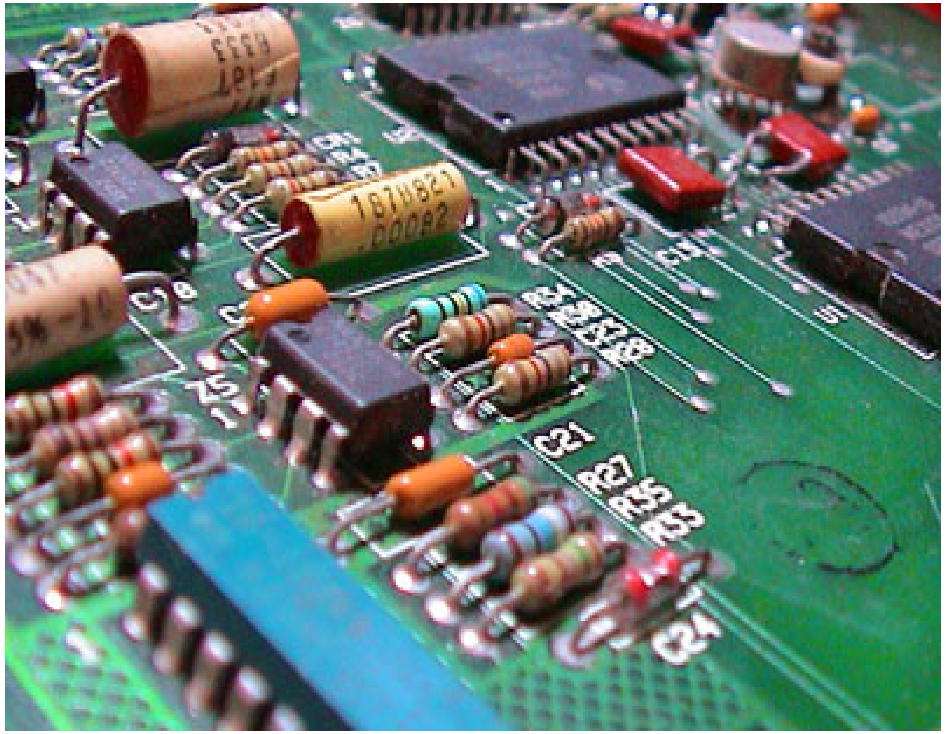


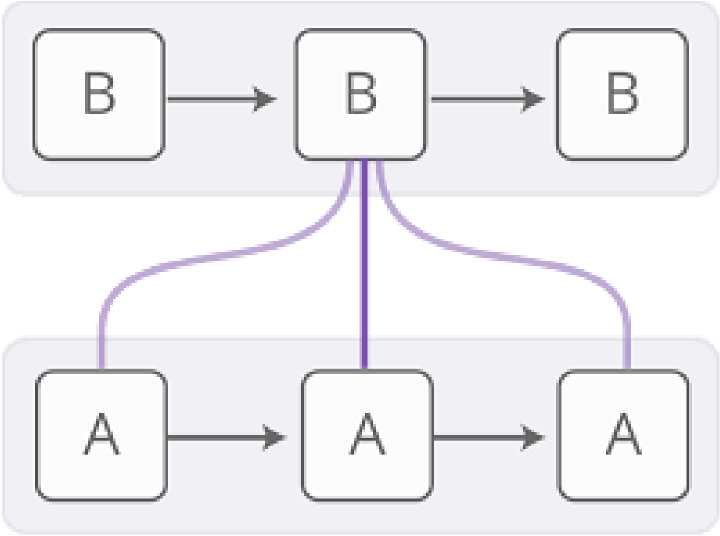
Photo credit: [Nicola Asuni]

Lecture 1: Sub-topics

- Reasoning as a prediction skill that can be learnt from data.
- Question answering as zero-shot learning.
- **Neural network operations for learning to reason:**
  - Attention & transformers.
  - Dynamic neural networks, conditional computation & differentiable programming.
- Reasoning as iterative representation refinement & query-driven program synthesis and execution.
  - Compositional attention networks.
  - Reasoning as Neural module networks.

Attentions: Picking up only what is needed at a step

- Need attention model to select or ignore certain **computations** or **inputs**
  - Can be "soft" (differentiable) or "hard" (requires RL)
  - Needed for selecting predicates in reasoning.
- Attention provides a short-cut → long-term dependencies
  - Needed for long chain of reasoning.
  - Also encourages sparsity if done right!



http://distill.pub/2016/augmented-rnns/

Fast weights | HyperNet – the multiplicative interaction

- Early ideas in early 1990s by Juergen Schmidhuber and collaborators.
- **Data-dependent weights** | Using a controller to generate weights of the main net.

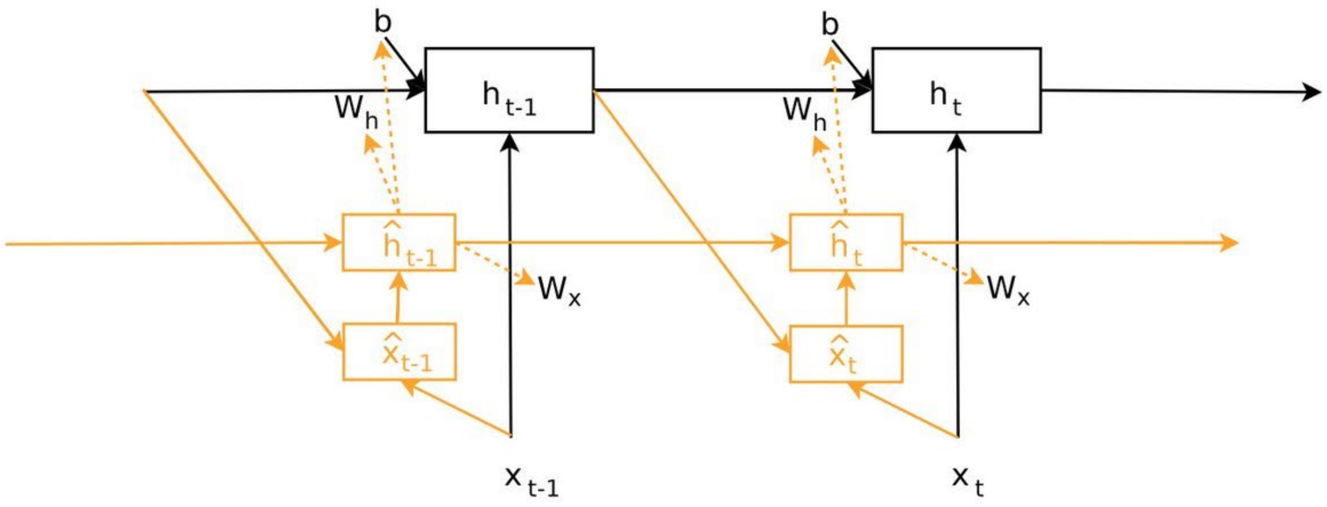
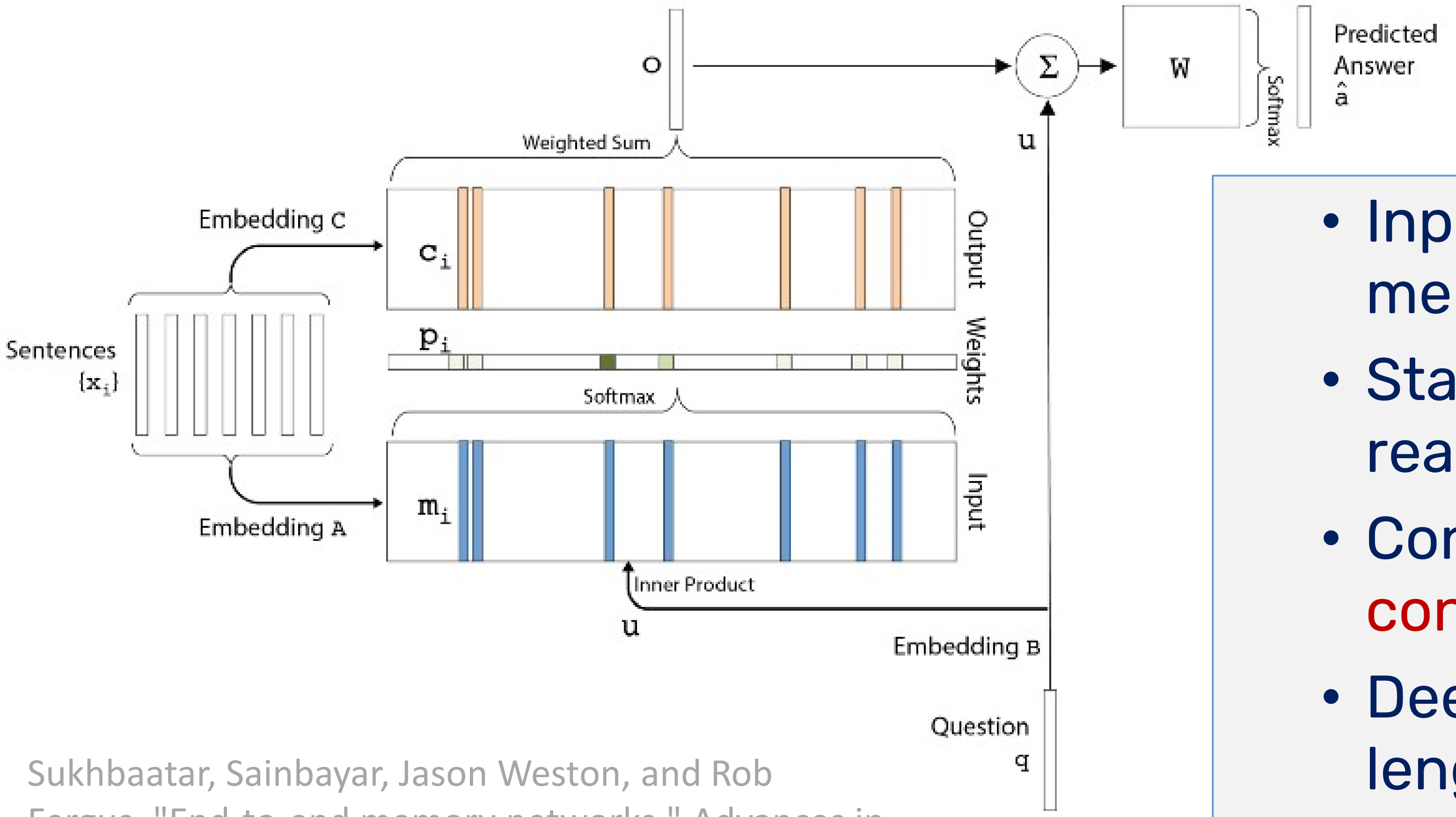


Figure: The HyperRNN system. The black system represents the main RNN while the orange system represents the weight-generating HyperRNN cell.

Ha, David, Andrew Dai, and Quoc V. Le. "Hypernetworks." *arXiv preprint arXiv:1609.09106* (2016).

Memory networks: Holding the data ready for inference



Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus. "End-to-end memory networks." Advances in neural information processing systems. 2015.

- Input is a set → Load into memory, which is NOT updated.
- State is a RNN with attention reading from inputs
- Concepts: Query, key and content + Content addressing.
- Deep models, but constant path length from input to output.
- Equivalent to a RNN with shared input set.

Transformers: Analogical reasoning through self-attention

State

$$A_h = \mathrm{Softmax}(\alpha Q_h K_h^\top) V_h$$

$$Q_h = \boldsymbol{W}_q X, K_h = \boldsymbol{W}_k X \text{ and } V_h = \boldsymbol{W}_v X$$

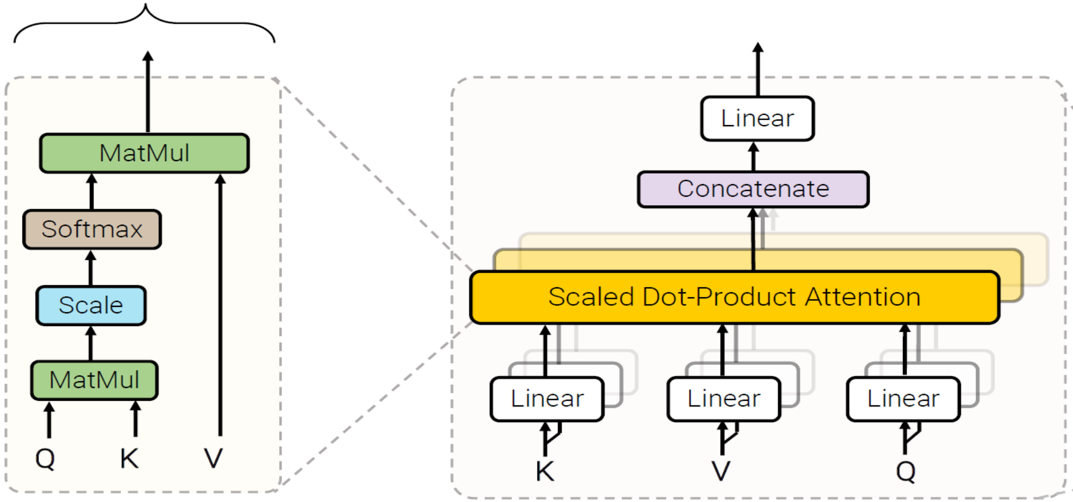
Query

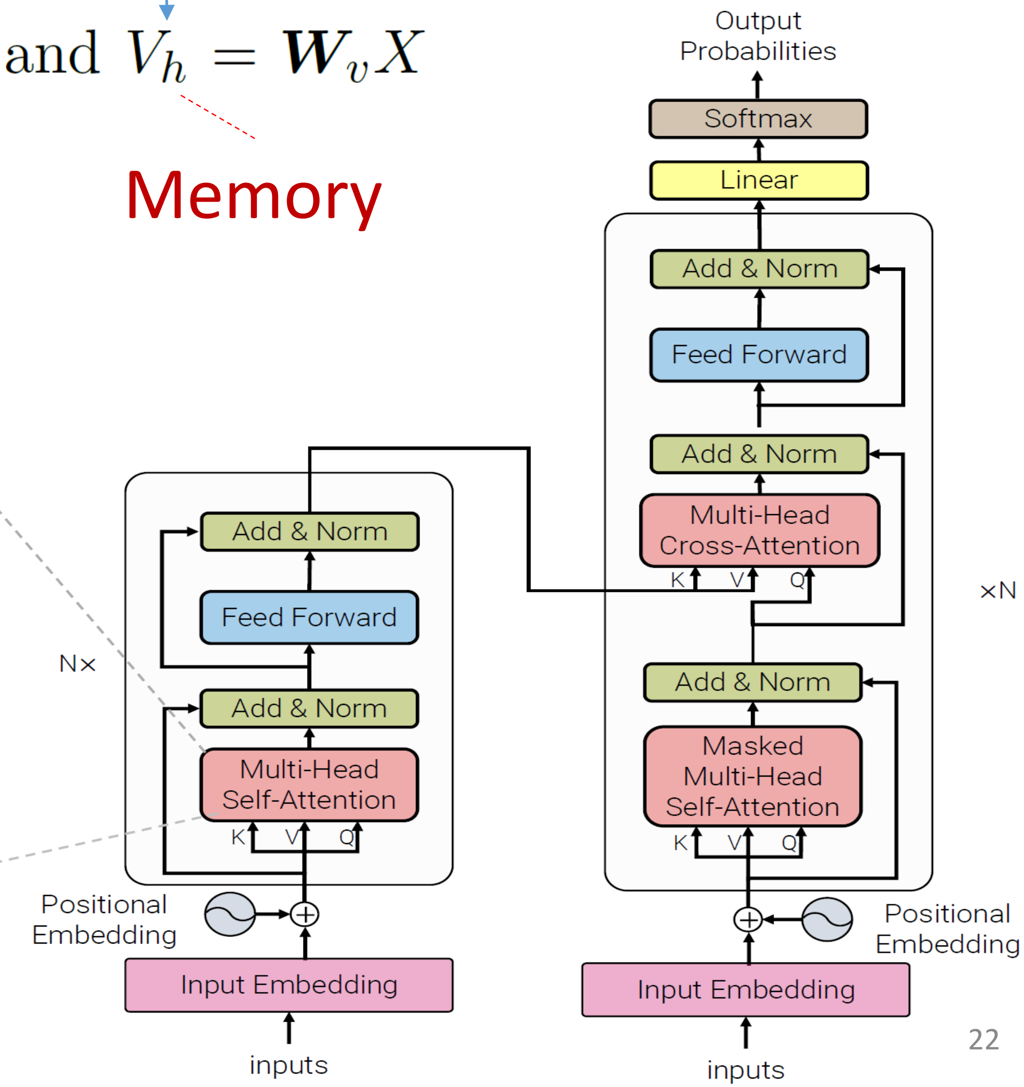
Key

Memory

Computational and Memory Complexity

$\mathcal{O}(n^2)$

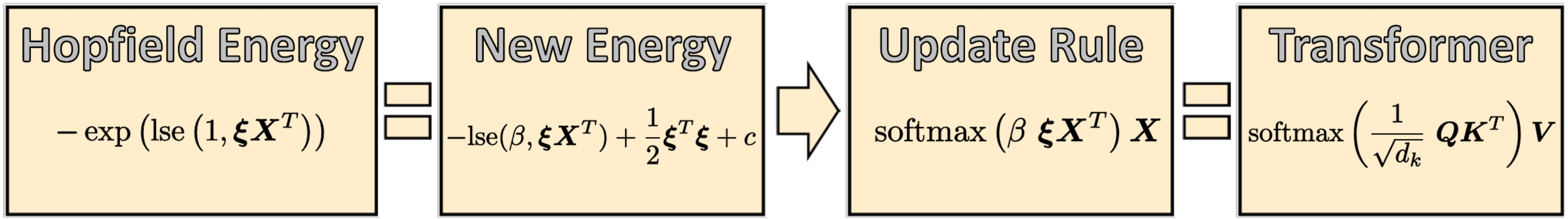




Tay, Yi, et al. "Efficient transformers: A survey." *arXiv preprint arXiv:2009.06732* (2020).

Transformer as implicit reasoning

- Recall: Reasoning as (free-) energy minimisation
  - The classic Belief Propagation algorithm is minimization algorithm of the Bethe free-energy!
- Transformer has relational, iterative state refinement makes it a great candidate for implicit relational reasoning.



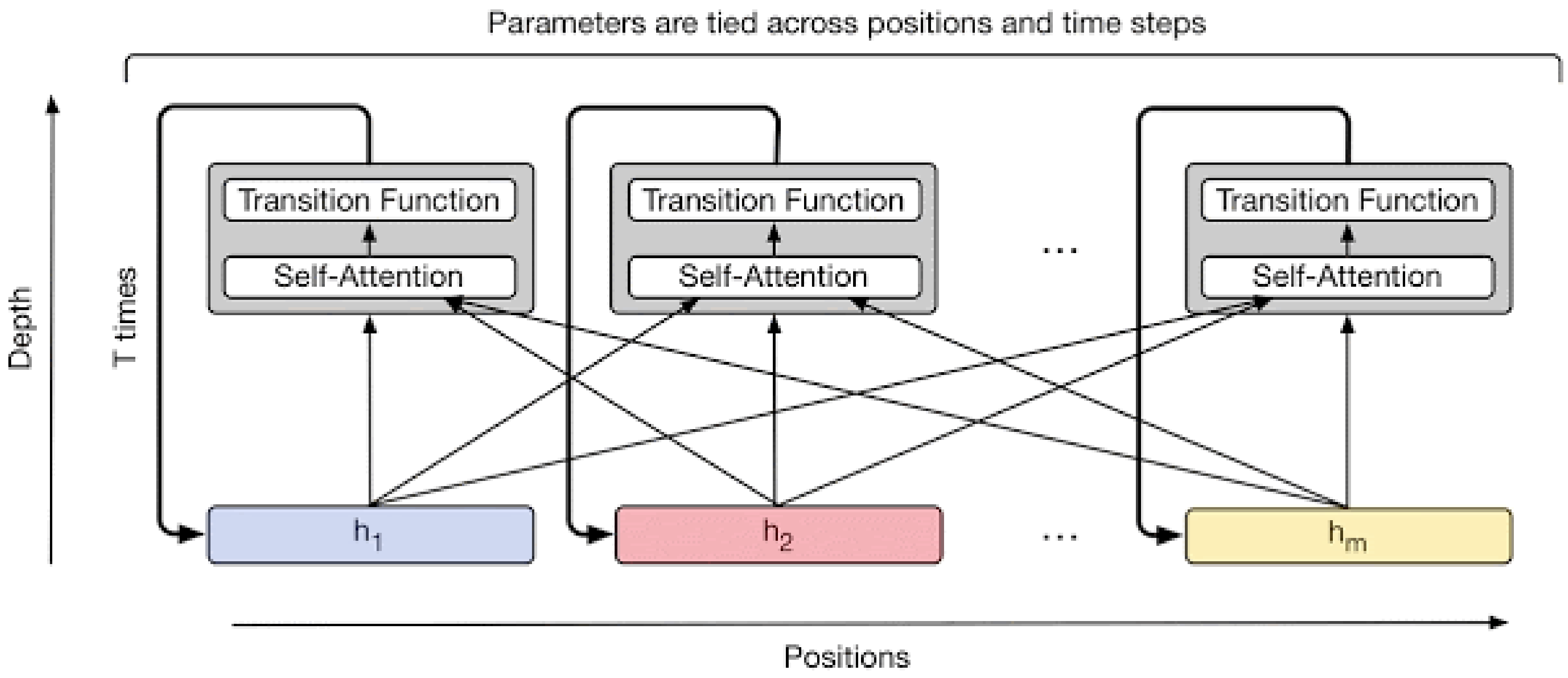
Ramsauer, Hubert, et al. "Hopfield networks is all you need." *arXiv preprint arXiv:2008.02217* (2020).

Transformer v.s. memory networks

- Memory network:
  - Attention to input set
  - One hidden state update at a time.
  - Final state integrate information of the set, conditioned on the query.
- Transformer:
  - Loading all inputs into working memory
  - Assigns one hidden state per input element.
  - All hidden states (including those from the query) to compute the answer.

Universal transformers

Dehghani, Mostafa, et al. "Universal Transformers." *International Conference on Learning Representations*. 2018.



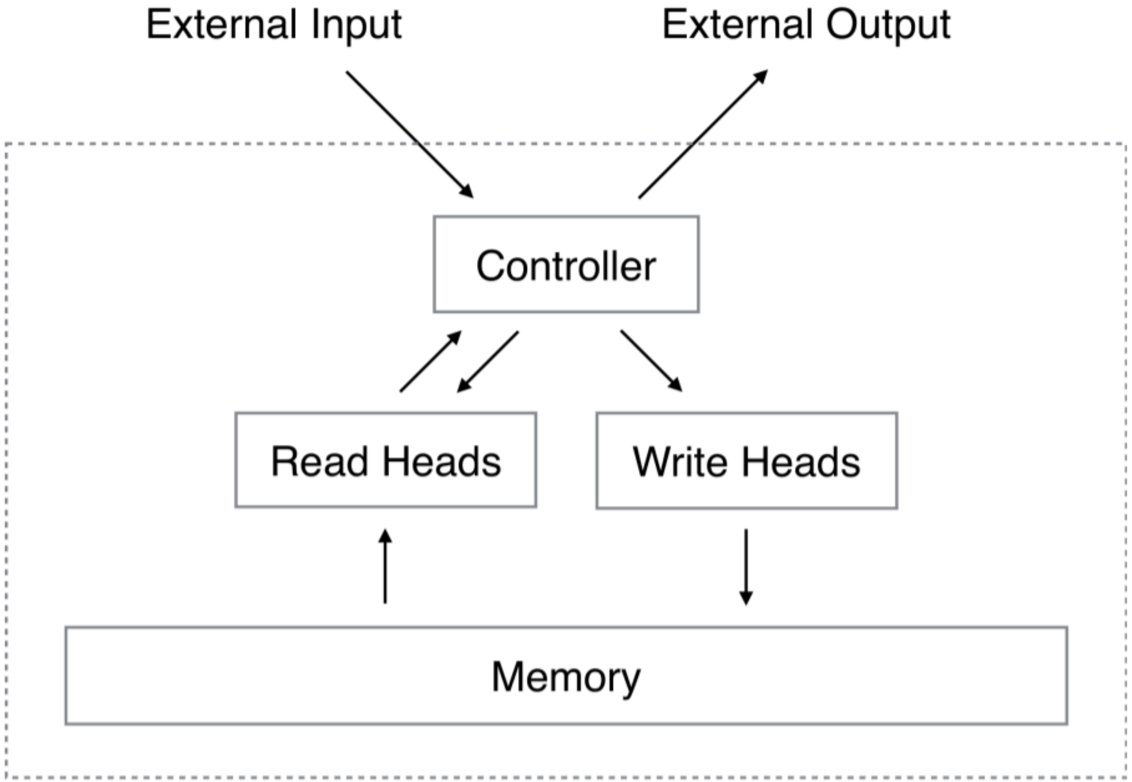
https://ai.googleblog.com/2018/08/moving-beyond-translation-with.html

Dynamic neural networks

- Memory-Augmented Neural Networks
- Modular program layout
- Program synthesis

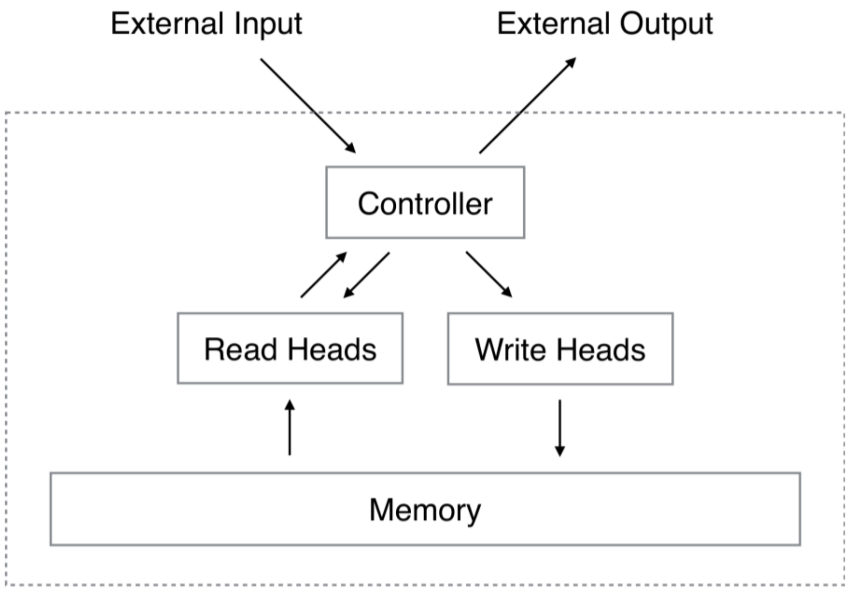
Neural Turing machine (NTM)
A memory-augmented neural network (MANN)

- A controller that takes input/output and talks to an external memory module.
- Memory has read/write operations.
- The main issue is where to write, and how to update the memory state.
- All operations are differentiable.



Source: rylanschaeffer.github.io

MANN for reasoning



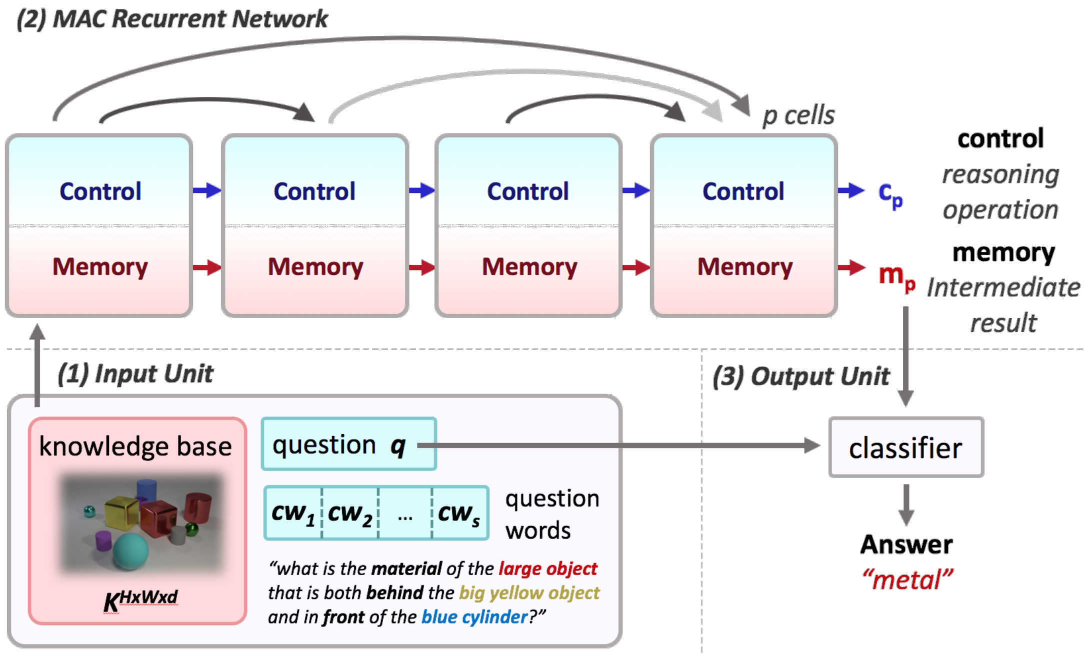
Source: rylanschaeffer.github.io

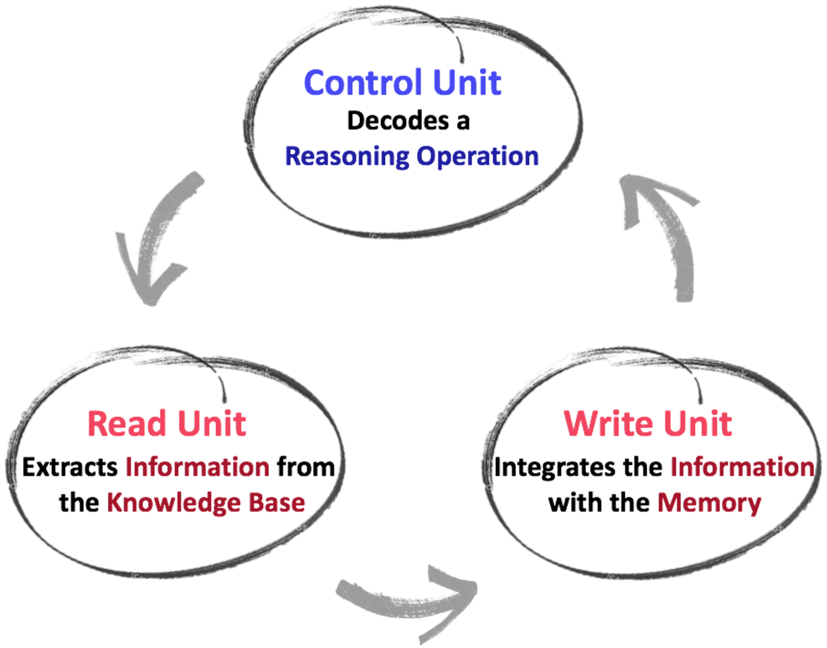
- Three steps:
  - Store data into memory
  - Read query, process sequentially, consult memory
  - Output answer
- Behind the scene:
  - Memory contains data & results of intermediate steps
  - LOGNet does the same, memory consists of object representations
- Drawbacks of current MANNs:
  - No memory of controllers → Less modularity and compositionality when query is complex
  - No memory of relations → Much harder to chain predicates.

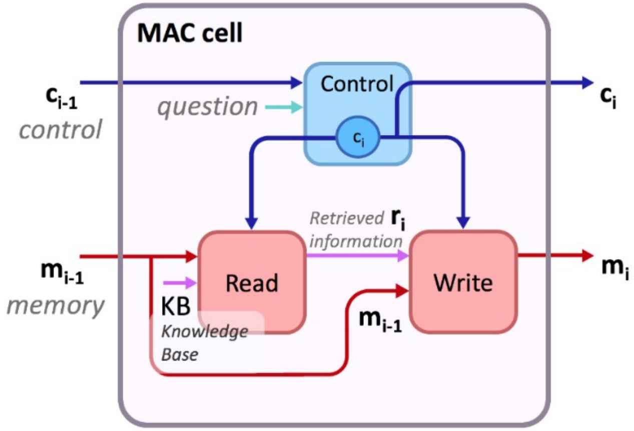
Lecture 1: Sub-topics

- Reasoning as a prediction skill that can be learnt from data.
  - Question answering as zero-shot learning.
- Neural network operations for learning to reason:
  - Attention & transformers.
  - Dynamic neural networks, conditional computation & differentiable programming.
- **Reasoning as iterative representation refinement & query-driven program synthesis and execution.**
  - Compositional attention networks.
  - Reasoning as Neural module networks.

MAC Net: Recurrent, iterative representation refinement





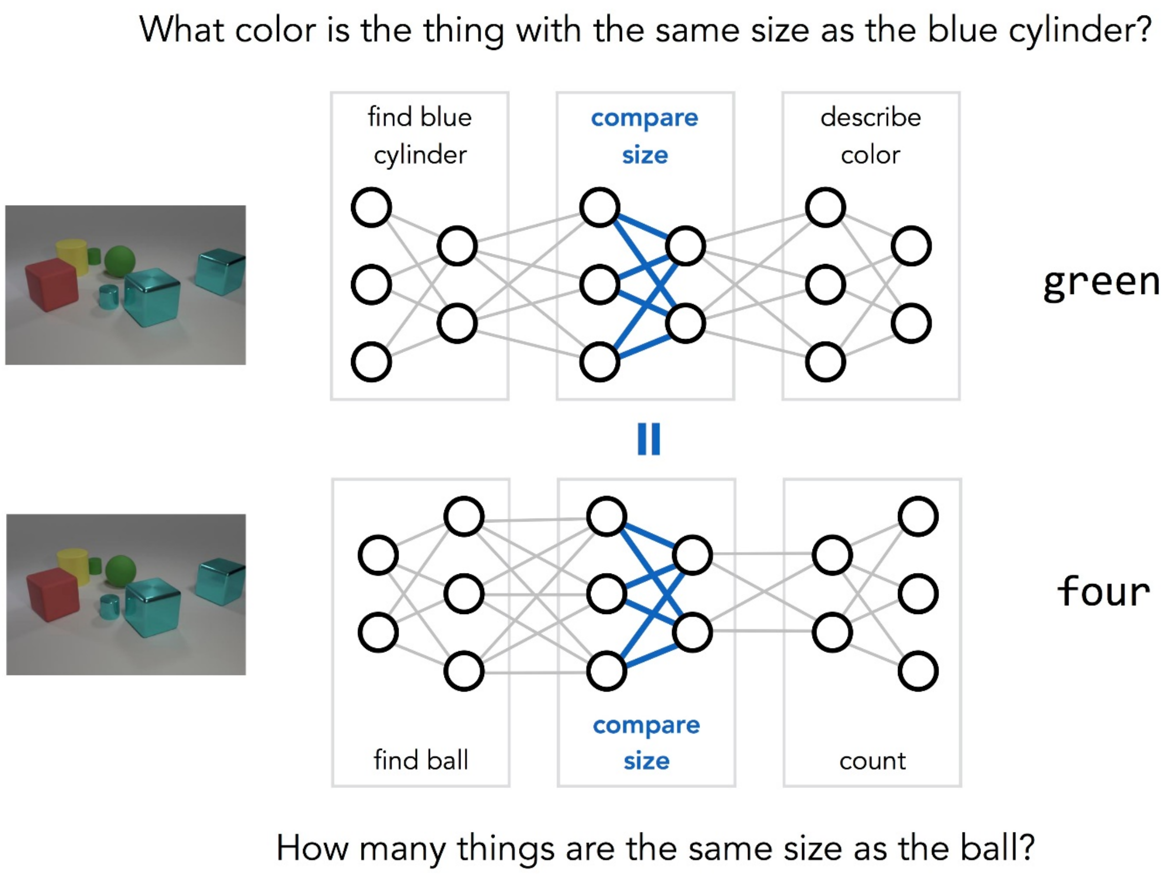


Hudson, Drew A., and Christopher D. Manning. "Compositional attention networks for machine reasoning." *ICLR* 2018.

Module networks

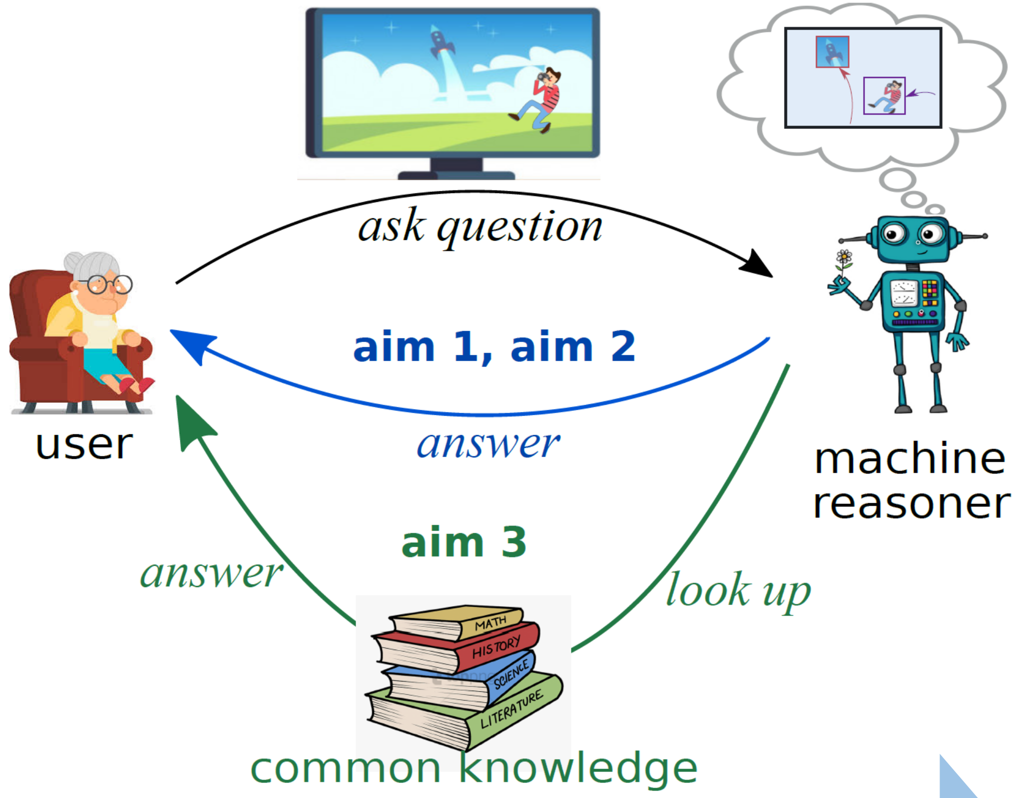
(reasoning by constructing and executing neural programs)

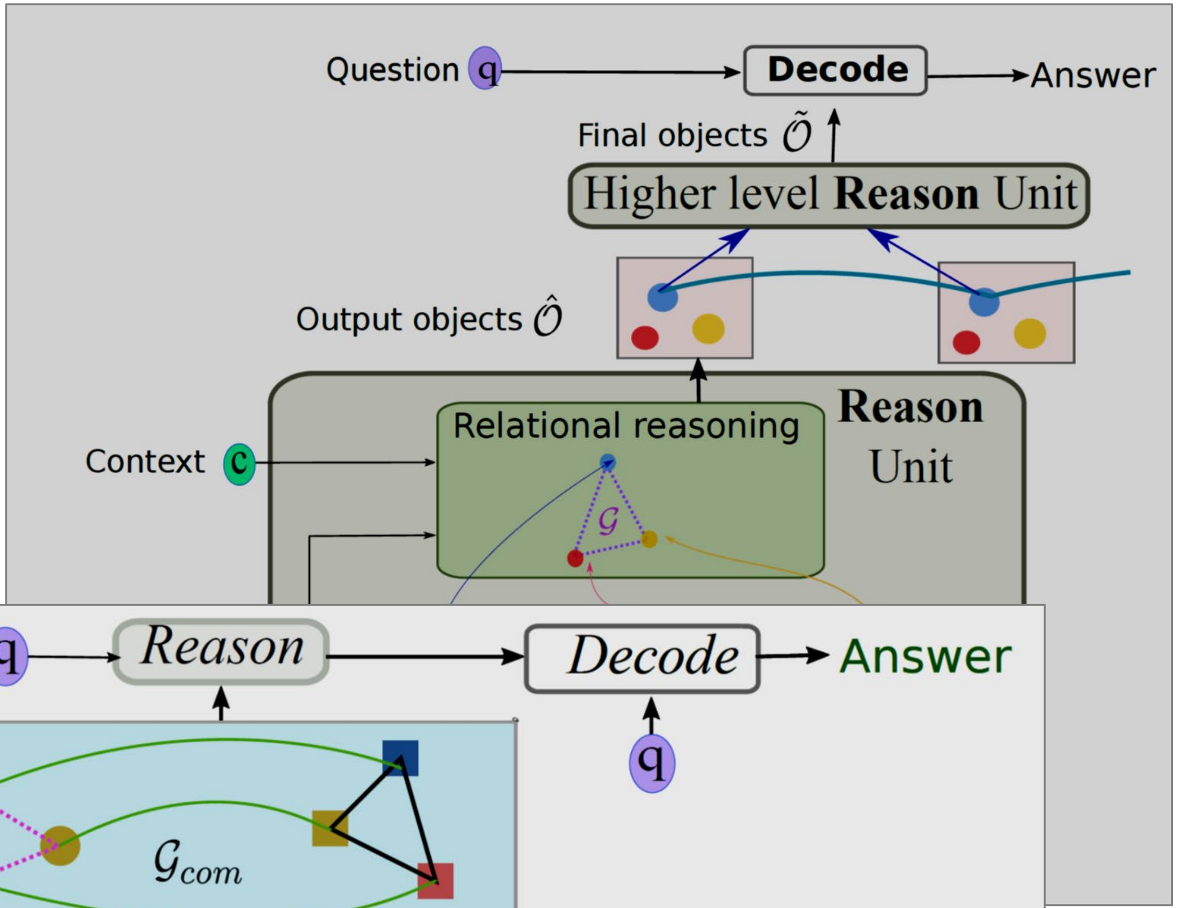
- Reasoning as laying out modules to reach an answer
- Composable neural architecture → question parsed as program (layout of modules)
- A module is a function (x → y), could be a sub-reasoning process ((x, q) → y).

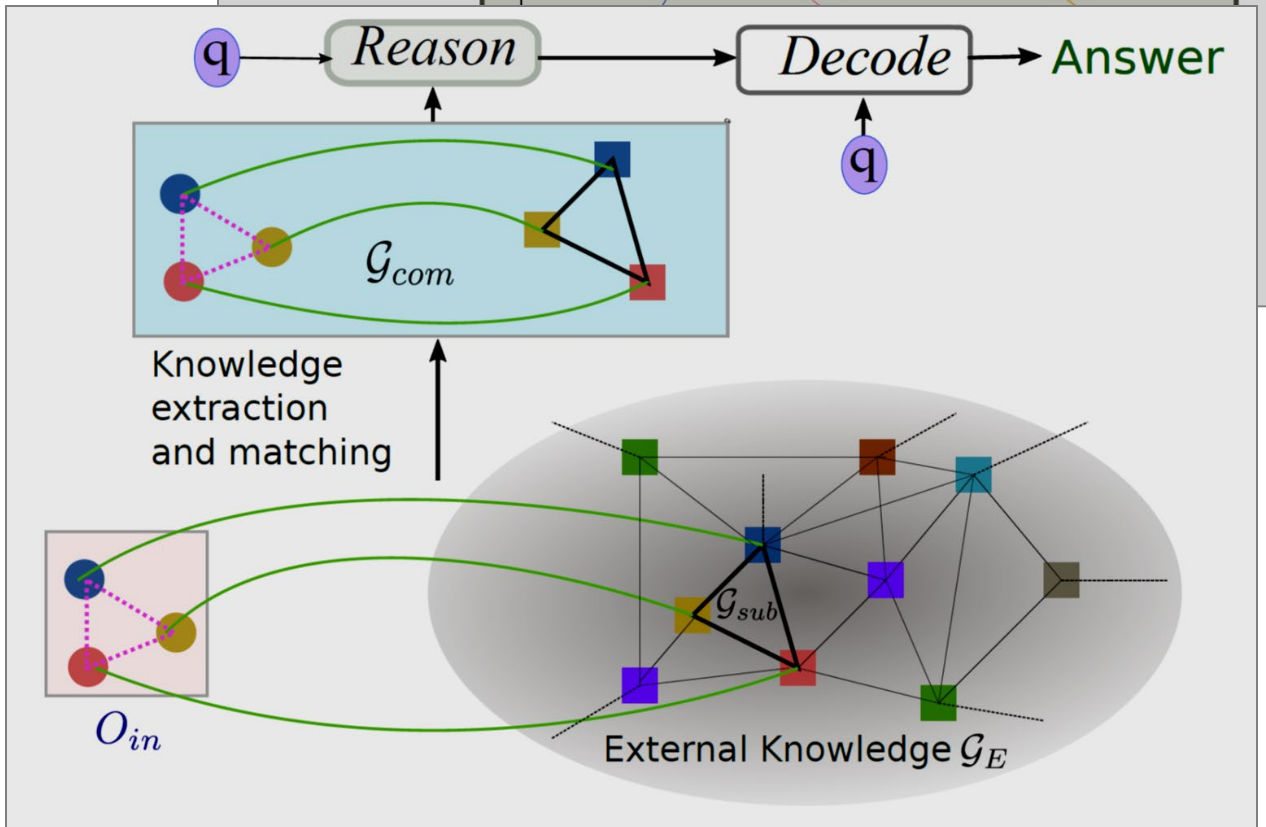


https://bair.berkeley.edu/blog/2017/06/20/learning-to-reason-with-neural-module-networks/

Example: A framework for visual reasoning







End of Lecture 1

https://neuralreasoning.github.io