Outline for today

- Presentation by *Ben Berman* on The Algebraic Mind (~45 min)
- Presentation by *Henry Lieberman* (~15 min)
- Discussion (~10 min)
- Presentation by Adam on
    - more modern neural implementations of variables
    - more modern connectionist language models (~20 min)
- More discussion (~20-30 min)

Things we should discuss today

- Breaking down the symbolic/connectionist debate into its sub-problems
    - Variabilized rules and syntactic generalization
    - Structured representations
    - Recursion
    - Individuals vs. kinds
- The historical context and contingencies around Marcus's 2001 book
    - Obsession circa late 1990s/early 2000s with multi-layer perceptrons and SRNs
        - And whether todays deep LSTMs are basically evolutions of the same concept
    - Prior framing around rules and variables (Frege, Tarski)
    - Prior work on linguistic vs. non-linguistic representations (Kosslyn)
- Apparent symbolic vs. connectionist positions of currently influential AI efforts
    - Hinton
    - Norvig
    - DeepMind
    - OpenAI
- Possible directions for development
    - Neural implementations of variables
    - Hybrid "neuro-symbolic" systems
    - Taught symbolic systems (e.g., Neural Turing Machine)
- Other topics you want to discuss?

What are “variables”?

Example #2: Modeling Syntax-Sensitive Rule-Based Decisions Using a DPAAN

As described in the introduction, cognitive modeling has traditionally assumed not only that syntax can be represented but that it can be recognized and manipulated as well. A classic example is the following:

John loves Mary. Mary loves Sam. John is jealous of who?

A system that understands the rules of jealousy would be able to correctly replace the “who” with “Sam.” Importantly, this system must generalize this rule only where appropriate allowing it to apply to all people. Below is a characterization of the jealously rule in a format using variables:

If ($\mathbf{X}_1$ loves $\mathbf{X}_2$) and ($\mathbf{X}_3$ loves $\mathbf{X}_4$) and ($\mathbf{X}_5$ is jealous of $\mathbf{X}_6$) and
($\mathbf{X}_1 = \mathbf{X}_5$) and ($\mathbf{X}_2 = \mathbf{X}_3$) and ($\mathbf{X}_1 \neq \mathbf{X}_4$)& ($\mathbf{X}_6$ = “*who*”)
Then ($\mathbf{X}_4 \rightarrow \mathbf{X}_6$)

In the above example the variables would be initialized as follows: $\mathbf{X}_1$ = “John,” $\mathbf{X}_2$ = “Mary,” $\mathbf{X}_3$ = “Mary,” $\mathbf{X}_4$ = “Sam,” $\mathbf{X}_5$ = “John,” $\mathbf{X}_6$ = “who.” This satisfies the if-clause leading to execution of the then-clause, namely the semantic contents of variable $\mathbf{X}_4$ will be transferred to $\mathbf{X}_6$. This transfer answers the question, signaling that John is in fact jealous of Sam.

HYPOTHESIS & THEORY ARTICLE
Front. Comput. Neurosci., 28 September 2012 | http://dx.doi.org/10.3389/fncom.2012.00073

Dynamically partitionable autoassociative networks as a solution to the neural binding problem

Kenneth J. Hayworth*
Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

What are “variables”?

Roles and Fillers

Rules depend on roles (variables) but not on filler contents

If we train a neural network to execute rules, want it to *generalize* to unseen fillers:

“Mary” --> “Hippopotamus”

One of Marcus’s main issues with the neural nets popular circa 2003 was their inability to fully generalize in this way

HYPOTHESIS & THEORY ARTICLE
Front. Comput. Neurosci., 28 September 2012 | http://dx.doi.org/10.3389/fncom.2012.00073

Dynamically partitionable autoassociative networks as a solution to the neural binding problem

Kenneth J. Hayworth*

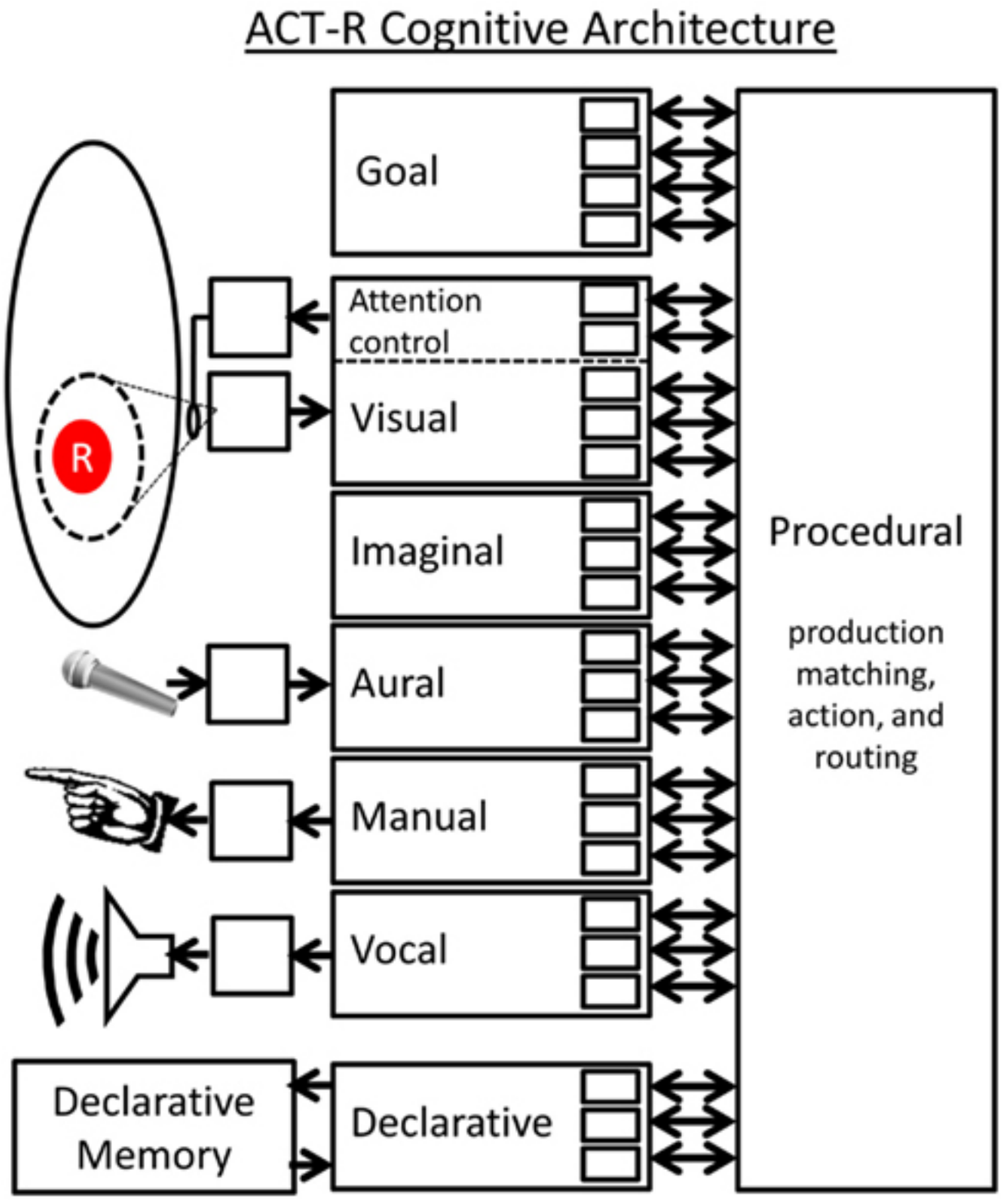
Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

What are “variables”?

ACT-R cognitive architecture
(hugely influential model)

Buffers
+
Parallel Rule Execution

If internal state matches execution conditions for a rule, update internal state accordingly



HYPOTHESIS & THEORY ARTICLE
Front. Comput. Neurosci., 28 September 2012 | http://dx.doi.org/10.3389/fncom.2012.00073

Dynamically partitionable autoassociative networks as a solution to the neural binding problem

Kenneth J. Hayworth*
Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

How to implement variables neurally?

Key elements for "anatomical variable binding":

1) Registers (roles) that can occupy a diverse set of states (fillers)
2) Copy and paste between registers
2) Equality detection between registers

HYPOTHESIS & THEORY ARTICLE

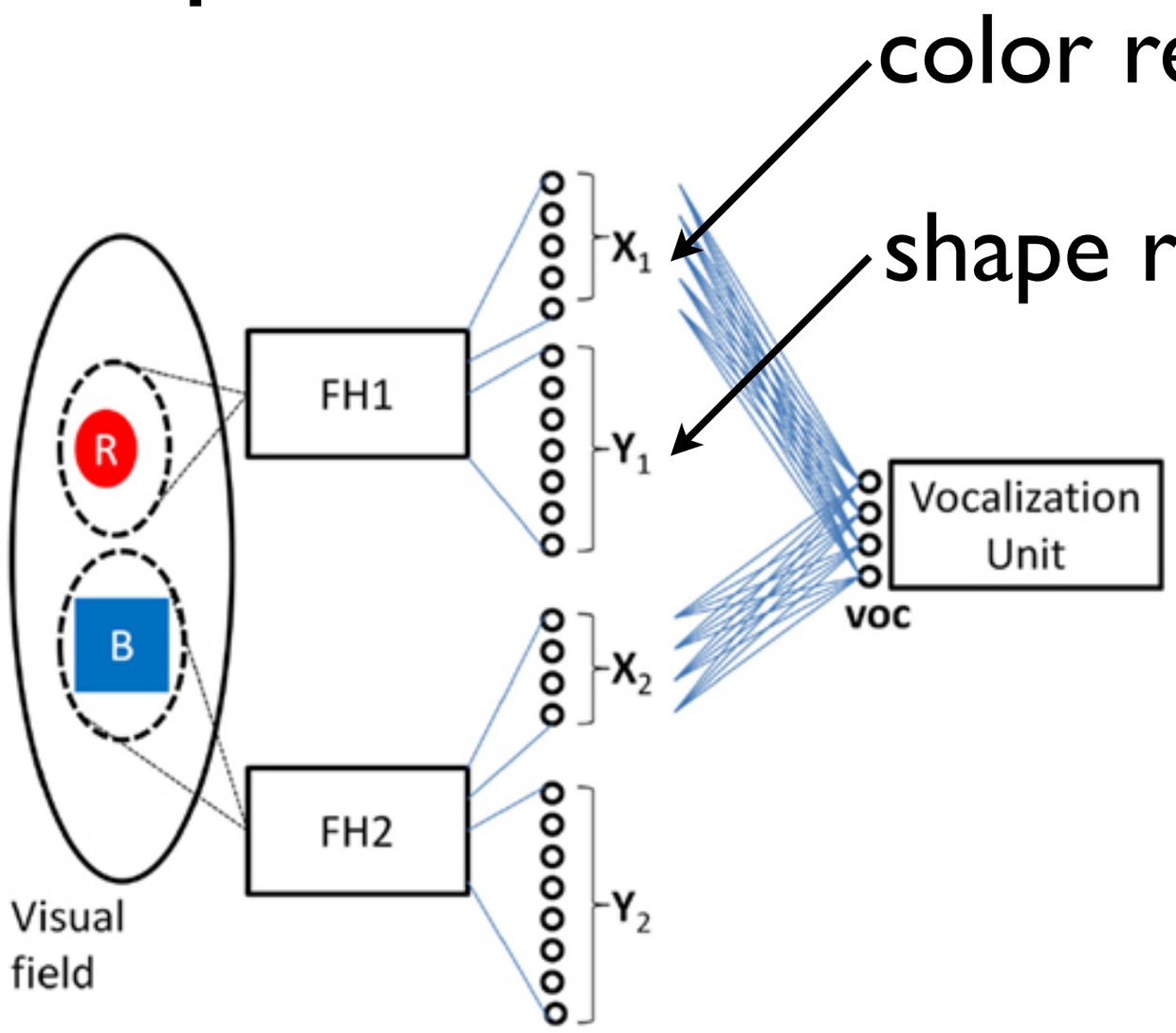
Front. Comput. Neurosci., 28 September 2012 | http://dx.doi.org/10.3389/fncom.2012.00073

Dynamically partitionable autoassociative networks as a solution to the neural binding problem

Kenneth J. Hayworth*

Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

How to implement variables neurally?



Q: Do the square and the circle have the same color?

Requires comparing contents between registers with different learned encodings

HYPOTHESIS & THEORY ARTICLE
Front. Comput. Neurosci., 28 September 2012 | http://dx.doi.org/10.3389/fncom.2012.00073

Dynamically partitionable autoassociative networks as a solution to the neural binding problem

Kenneth J. Hayworth*
Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

How to implement variables neurally?

Key elements for "anatomical variable binding":

1) Registers (roles) that can occupy a diverse set of states (fillers)
   - Register = Groups of neurons
   - State = Stable activity pattern (attractor) of group of neurons
2) Copy and paste between registers
   - Global attractor states linking registers
   - Turn off connections between registers to partition them
   - Turn on connections between registers to force occupation of same state
2) Equality detection between registers
   - Uses a separate network that computes an "energy function" of the neural states

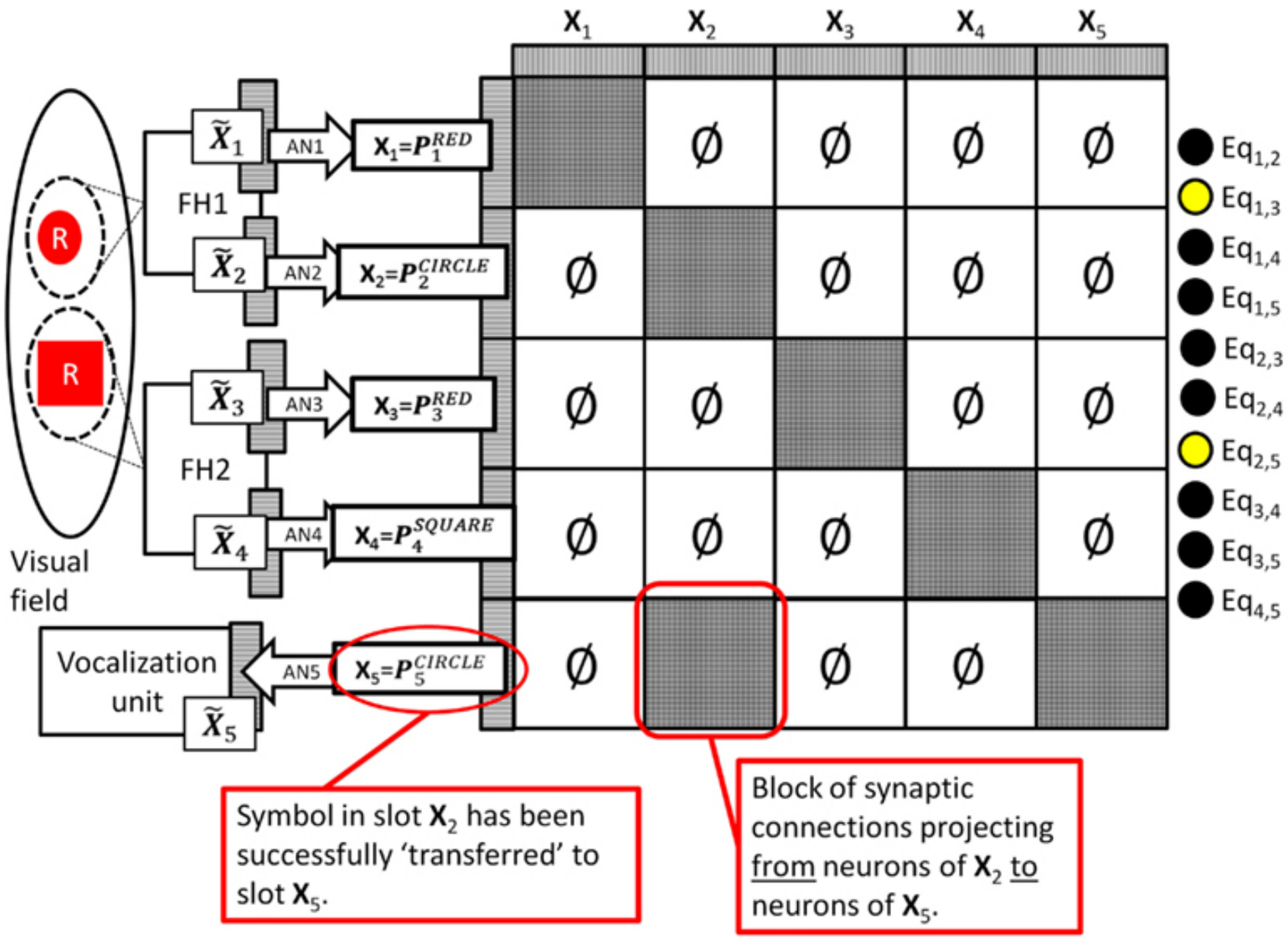
HYPOTHESIS & THEORY ARTICLE
Front. Comput. Neurosci., 28 September 2012 | http://dx.doi.org/10.3389/fncom.2012.00073

Dynamically partitionable autoassociative networks as a solution to the neural binding problem

Kenneth J. Hayworth*

Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

How to implement variables neurally?



HYPOTHESIS & THEORY ARTICLE
Front. Comput. Neurosci., 28 September 2012 | http://dx.doi.org/10.3389/fncom.2012.00073

Dynamically partitionable autoassociative networks as a solution to the neural binding problem

Kenneth J. Hayworth*
Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

How to implement variables neurally?

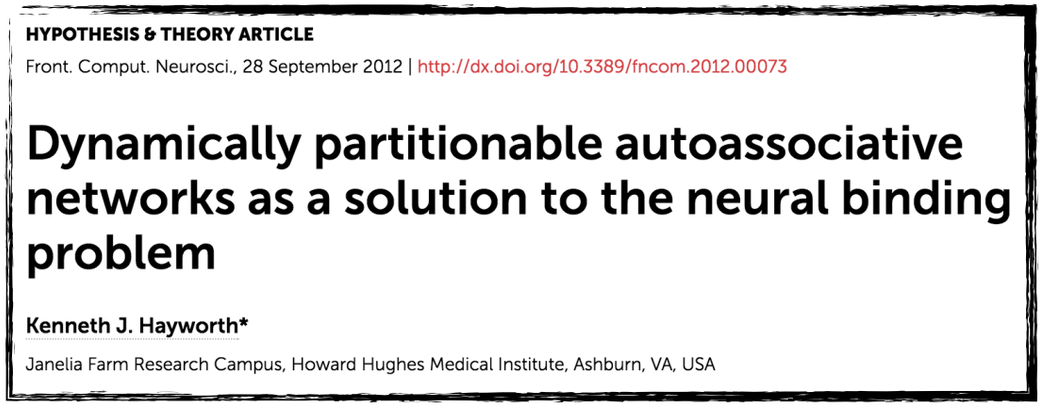
Notes about DPAAN model:

Made of neurons but not a multilayer perceptron or SRN

This machinery uses *learned* symbols arising from hierarchical pattern recognition (e.g., deep nets)

The neural variable-manipulating machinery is self-organized (by Hebbian learning, to form the attractor states) but is not "trained" based on data in traditional sense

Similar to Marcus's suggestion of register operations



HYPOTHESIS & THEORY ARTICLE

Front. Comput. Neurosci., 28 September 2012 | http://dx.doi.org/10.3389/fncom.2012.00073

Dynamically partitionable autoassociative networks as a solution to the neural binding problem

Kenneth J. Hayworth*

Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

Modern connectionist language models

Represent words, phrases as high-dimensional (e.g., 500-dim) vectors

Vectors, in turn, are represented as **distributed** neural activity patterns in a group of "neurons"

Tensor operations on vectors for words lead to vectors for phrases

See also:

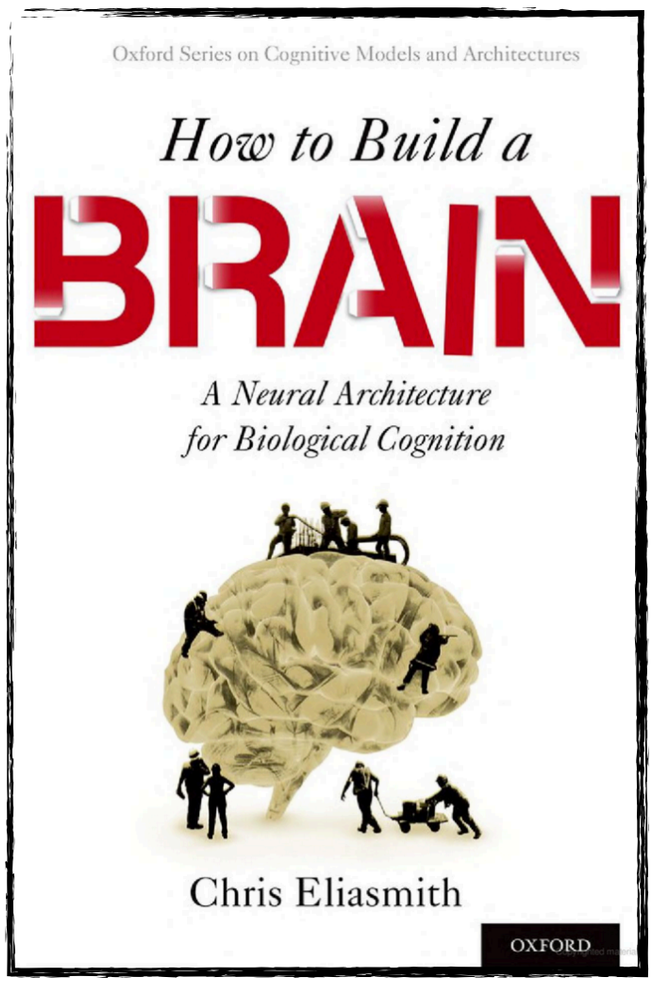
-"Holographic reduced representations"
-Recent ML work by Richard Socher

Can de-code the phrase vectors to extract the individual word vectors

Modern connectionist language models

A Neurally Plausible Encoding of Word Order Information into a Semantic Vector Space

Peter Blouw (pblouw@uwaterloo.ca)
Chris Eliasmith (celiasmith@uwaterloo.ca)
Center for Theoretical Neuroscience, University of Waterloo
Waterloo, ON N2L3G1 Canada

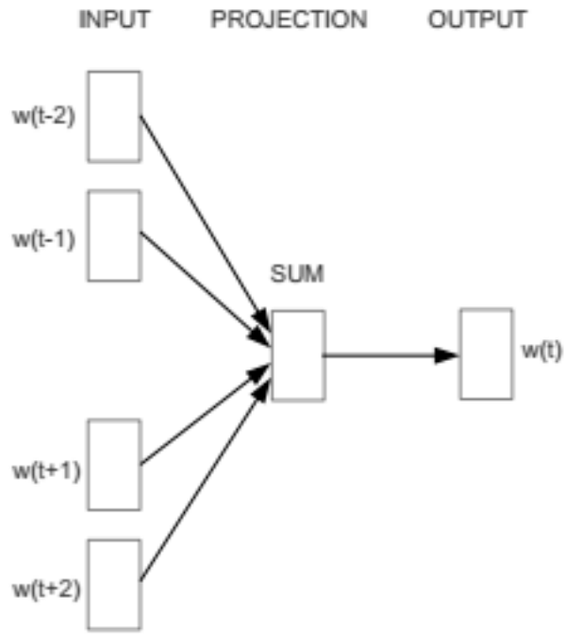


Modern connectionist language models

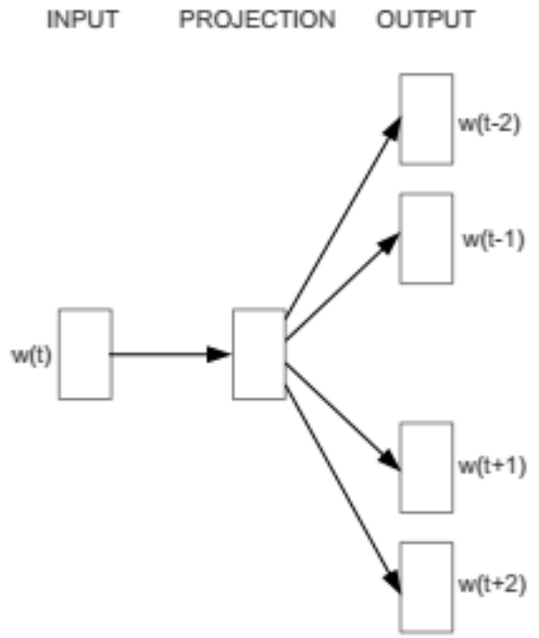
word2vec: finding "semantic" vector representations of words

train neural net to predict a given word, represented as a vector, from its surrounding words

leads to good vectors occupying a "semantic space"



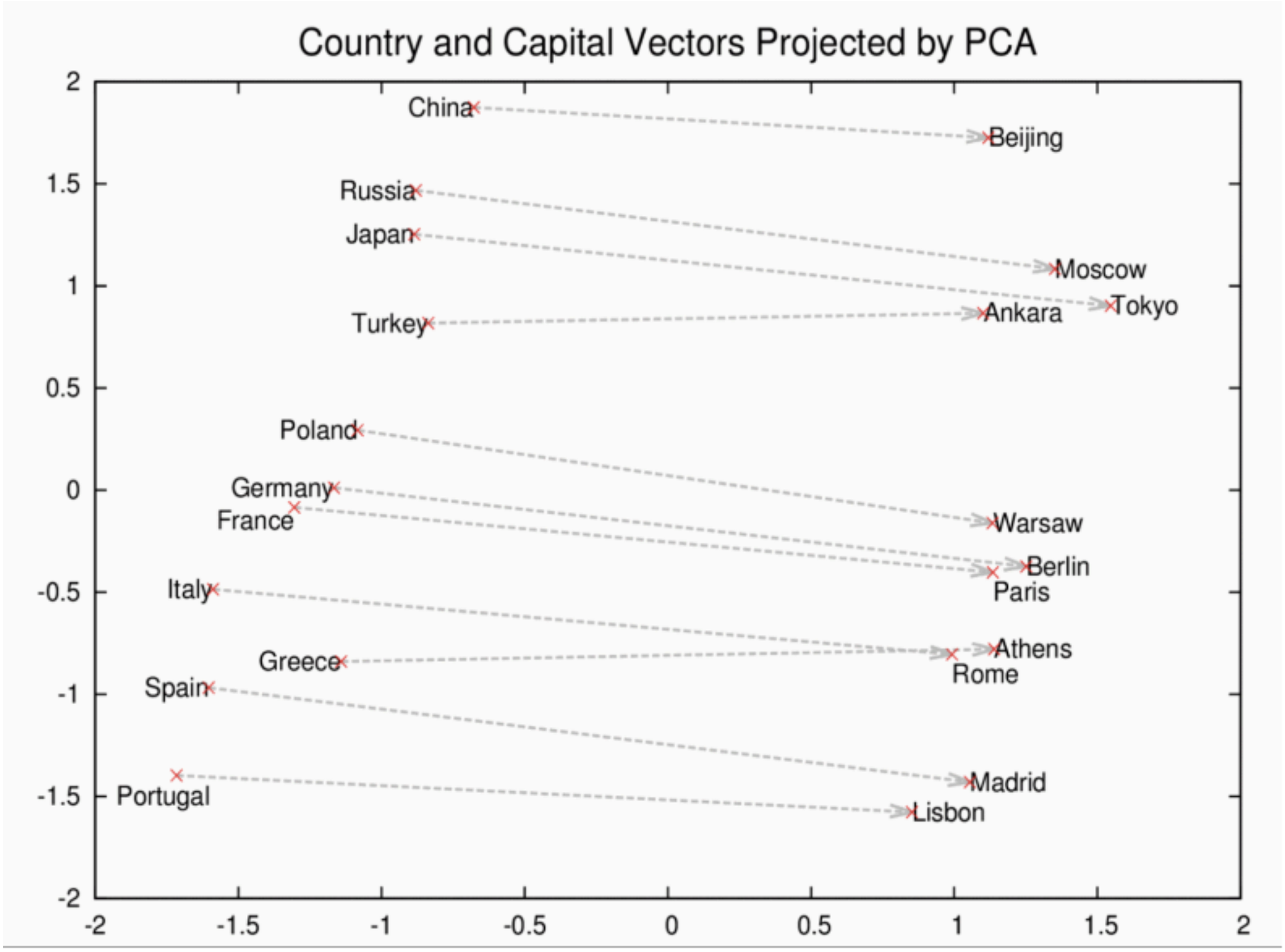
CBOW



Skip-gram

Modern connectionist language models

word2vec: finding "semantic" vector representations of words



Modern connectionist language models

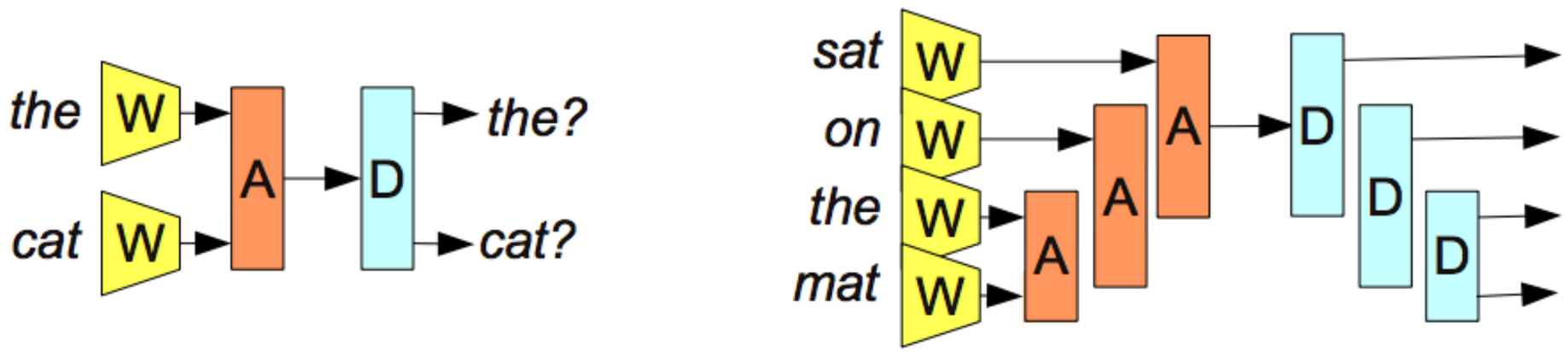


Figure 10 – Navigating intermediate representations with the dissociation module.

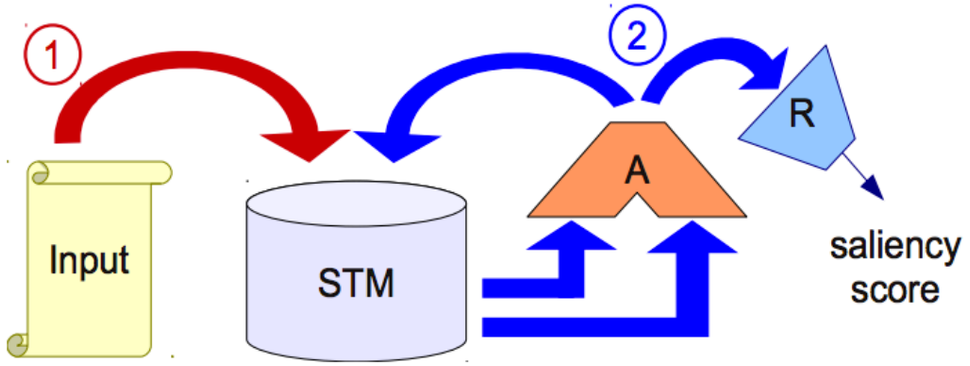


Figure 12 – The short-term memory (STM) holds a collection of representation vectors. The two possible actions are (1) inserting a new representation vector into the STM, and (2) replacing two vectors from the STM by the output of the association module.

http://arxiv.org/abs/1102.1808

From Machine Learning to Machine Reasoning

Léon Bottou

2/8/2011

Modern connectionist language models

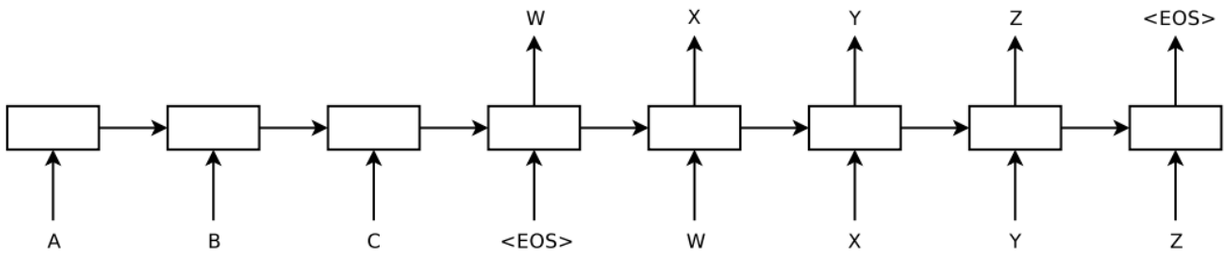
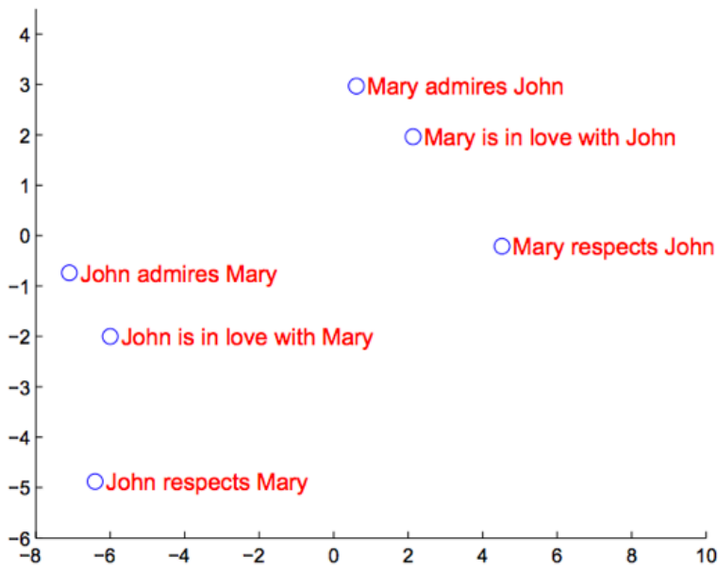


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

3.8 Model Analysis



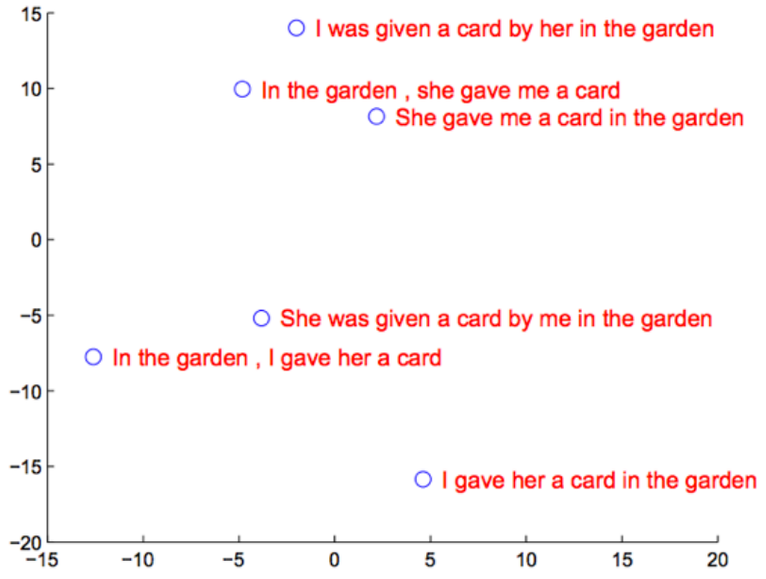


Figure 2: The figure shows a 2-dimensional PCA projection of the LSTM hidden states that are obtained after processing the phrases in the figures. The phrases are clustered by meaning, which in these examples is primarily a function of word order, which would be difficult to capture with a bag-of-words model. Notice that both clusters have similar internal structure.

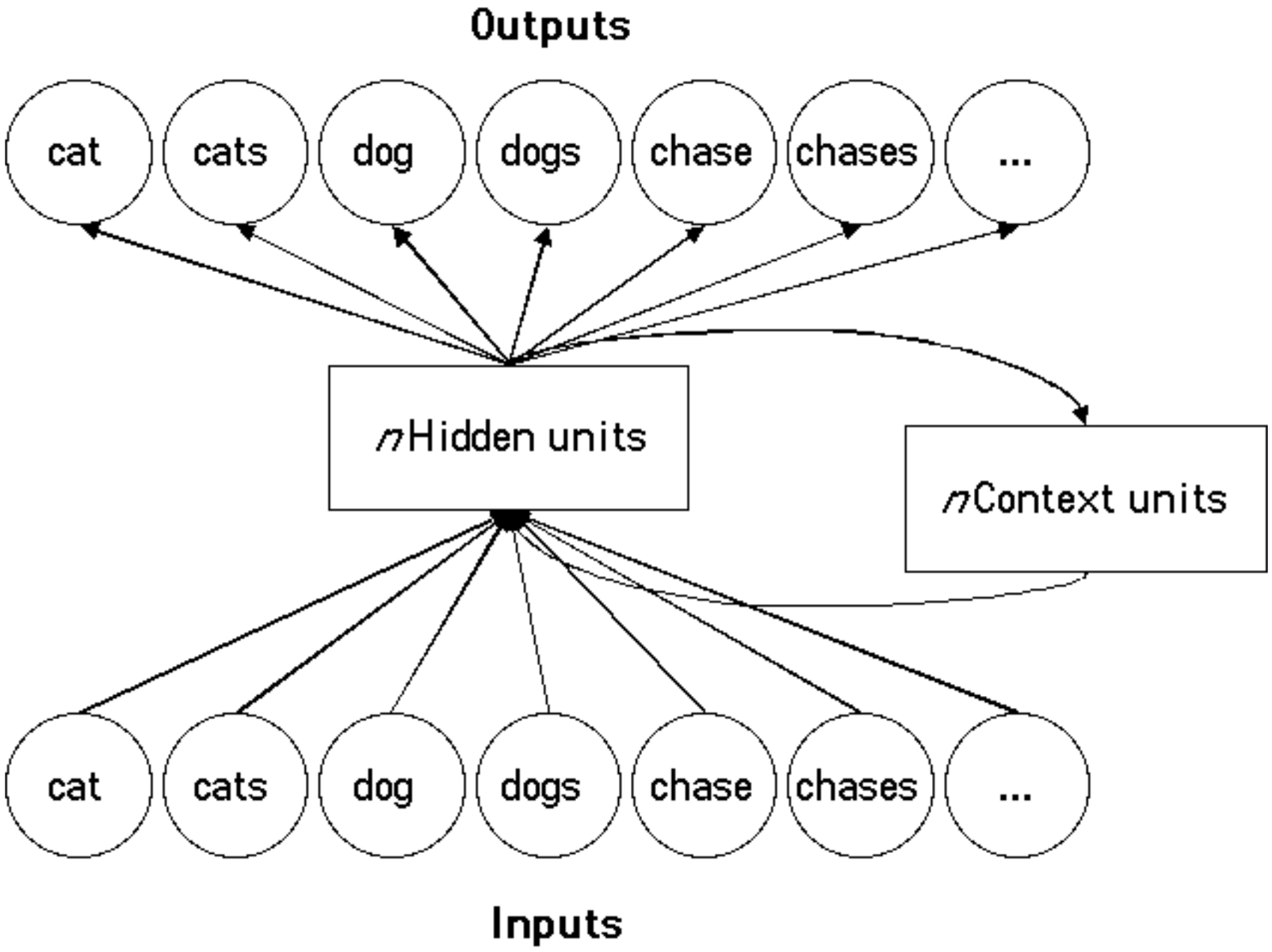
Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Modern connectionist language models



Compare with:

Elman Simple Recurrent Network cited by Marcus 2001

Key difference is that the networks learn distributed representations for the words themselves?
Also: LSTM cells rather than sigmoid neurons, many layers, stores state across many cycles

hmm... have they finally made peace? or not?

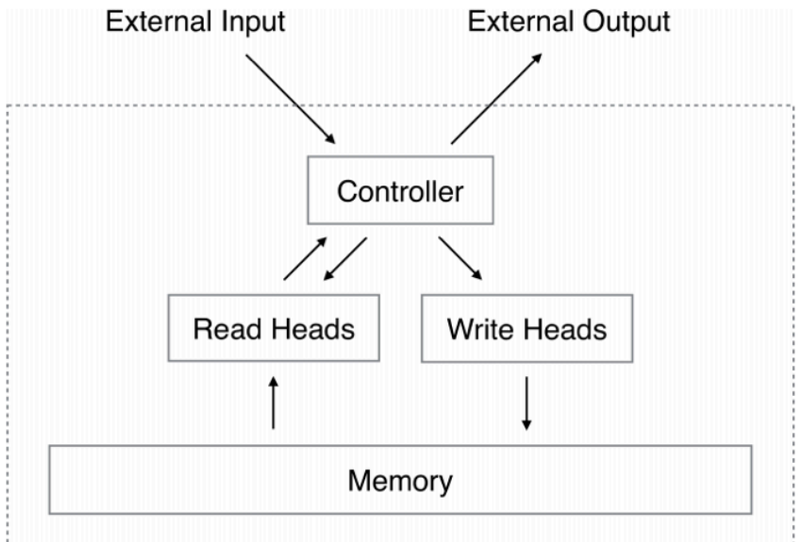


Figure 1: Neural Turing Machine Architecture. During each update cycle, the controller network receives inputs from an external environment and emits outputs in response. It also reads to and writes from a memory matrix via a set of parallel read and write heads. The dashed line indicates the division between the NTM circuit and the outside world.

Neural Turing Machines

Alex Graves, Greg Wayne, Ivo Danihelka

(Submitted on 20 Oct 2014 (v1), last revised 10 Dec 2014 (this version, v2))

We extend the capabilities of neural networks by coupling them to external memory resources, which they can interact with by attentional processes. The combined system is analogous to a Turing Machine or Von Neumann architecture but is differentiable end-to-end, allowing it to be efficiently trained with gradient descent. Preliminary results demonstrate that Neural Turing Machines can infer simple algorithms such as copying, sorting, and associative recall from input and output examples.

Subjects: **Neural and Evolutionary Computing (cs.NE)**
Cite as: **arXiv:1410.5401 [cs.NE]**
(or **arXiv:1410.5401v2 [cs.NE]** for this version)

A "taught symbolic system"?

hmm... have they finally made peace? or not?

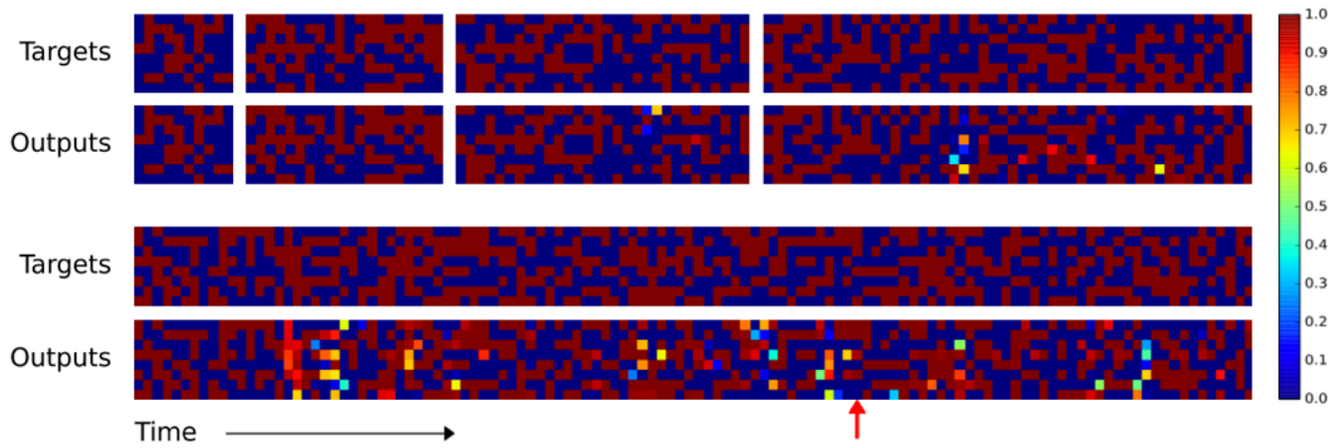


Figure 4: NTM Generalisation on the Copy Task. The four pairs of plots in the top row depict network outputs and corresponding copy targets for test sequences of length 10, 20, 30, and 50, respectively. The plots in the bottom row are for a length 120 sequence. The network was only trained on sequences of up to length 20. The first four sequences are reproduced with high confidence and very few mistakes. The longest one has a few more local errors and one global error: at the point indicated by the red arrow at the bottom, a single vector is duplicated, pushing all subsequent vectors one step back. Despite being subjectively close to a correct copy, this leads to a high loss.

Neural Turing Machines

Alex Graves, Greg Wayne, Ivo Danihelka

(Submitted on 20 Oct 2014 (v1), last revised 10 Dec 2014 (this version, v2))

We extend the capabilities of neural networks by coupling them to external memory resources, which they can interact with by attentional processes. The combined system is analogous to a Turing Machine or Von Neumann architecture but is differentiable end-to-end, allowing it to be efficiently trained with gradient descent. Preliminary results demonstrate that Neural Turing Machines can infer simple algorithms such as copying, sorting, and associative recall from input and output examples.

Subjects: **Neural and Evolutionary Computing (cs.NE)**
Cite as: **arXiv:1410.5401 [cs.NE]**
(or **arXiv:1410.5401v2 [cs.NE]** for this version)

hmm... have they finally made peace? or not?

connectionist?

NATURE | ARTICLE

日本語要約

Mastering the game of Go with deep neural networks and tree search

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel & Demis Hassabis

Affiliations | Contributions | Corresponding authors

symbolic?

Things we should discuss today

- Breaking down the symbolic/connectionist debate into its sub-problems
    - Variabilized rules and syntactic generalization
    - Structured representations
    - Recursion
    - Individuals vs. kinds
- The historical context and contingencies around Marcus's 2001 book
    - Obsession circa late 1990s/early 2000s with multi-layer perceptrons and SRNs
        - And whether todays deep LSTMs are basically evolutions of the same concept
    - Prior framing around rules and variables (Frege, Tarski)
    - Prior work on linguistic vs. non-linguistic representations (Kosslyn)
- Apparent symbolic vs. connectionist positions of currently influential AI efforts
    - Hinton
    - Norvig
    - DeepMind
    - OpenAI
- Possible directions for development
    - Neural implementations of variables
    - Hybrid "neuro-symbolic" systems
    - Taught symbolic systems (e.g., Neural Turing Machine)
- Other topics you want to discuss?