

Abstract rule representations in a bilinear model

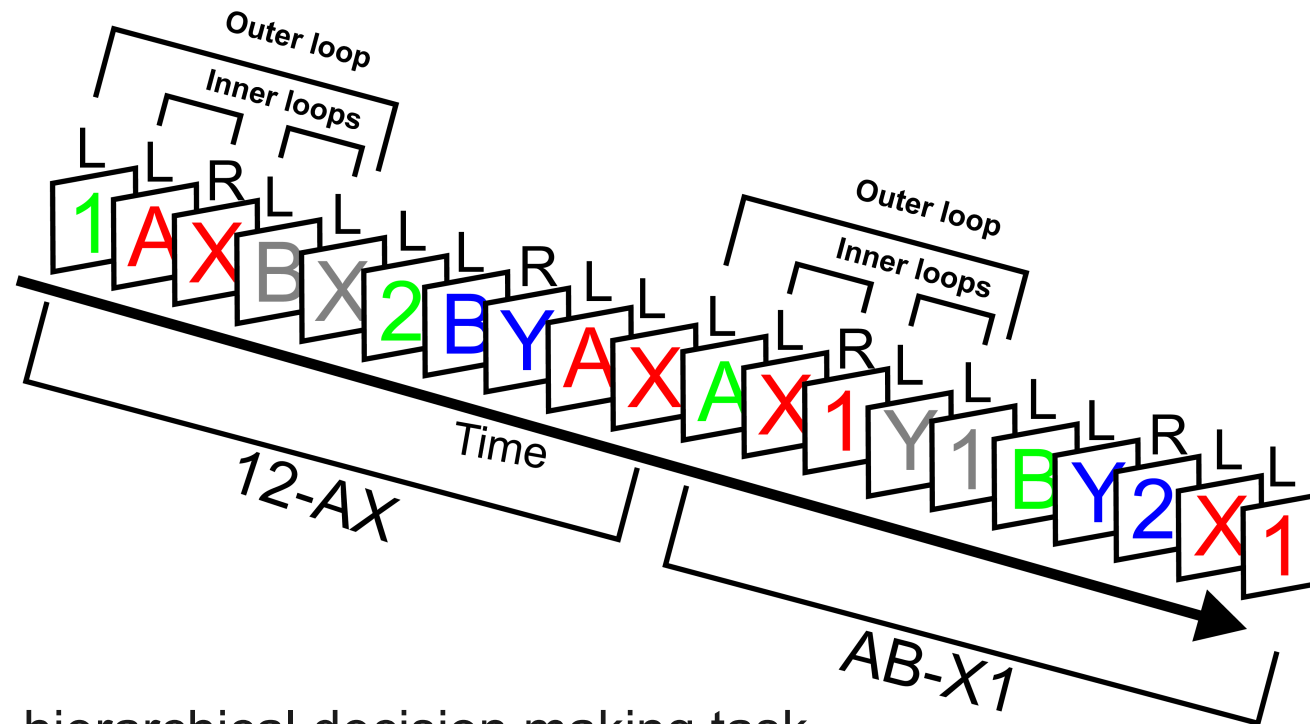
Computational and Systems Neuroscience Conference, 2009

Kai Krueger and Peter Dayan
Gatsby Computational Neuroscience Unit

Introduction

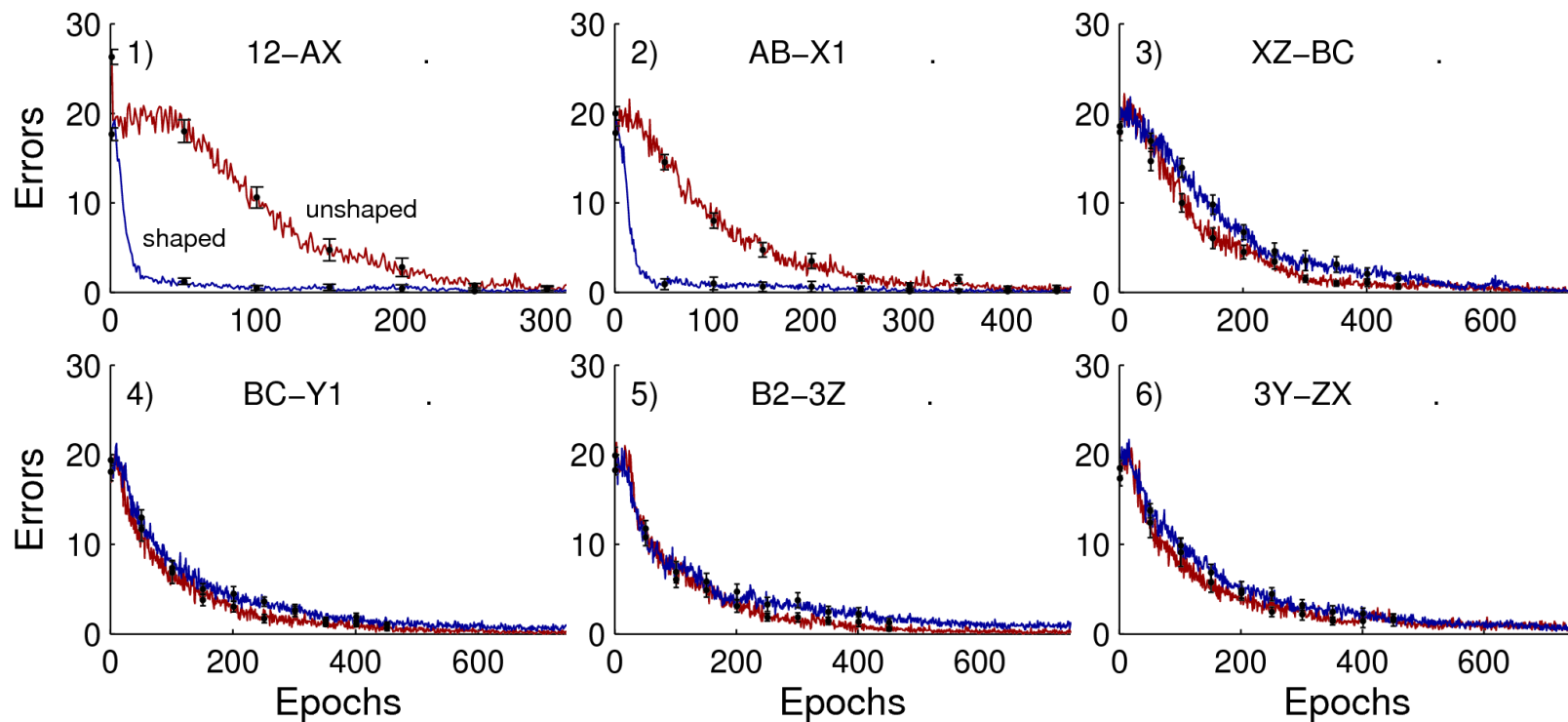
- A key aspect of **cognitive flexibility** is **abstraction**. i.e. the ability to separate and independently vary general rules and specific instantiations
- Delayed match to sample
 - rule: match a sequence to a target
 - instantiation: first presentation in the sequence
- Sequence categories:
 - rule: ABAB / AABB
 - instantiation: push – pull motion
- Poses a challenge for standard neural network models
 - stimulus **identities** are typically **encoded** in rule **weights**.
- Need rules (network weights) operating on rapidly updateable **variables**
 - adds the layer of abstraction
- Model a task with constant rules, but changing stimulus mapping

Generalised 12-AX



- Sequential, hierarchical decision making task
- Rules:
 - **outer** loop: present one of two possible “context” markers
 - **inner** loop: pair of stimuli randomly drawn from alphabet
 - each context has one target loop to which a respond to.
- **Abstract rules** and **concrete stimuli** are independent
 - keep rules fixed and switch instantiations of stimuli
- 12-AX task (*Frank 01 / O'Reilly 05*) is a specific case where “1” and “2” represent context and “AX”, “BY” the respective target sequences.

Learning in a recurrent Neural Network



- Learning and abstracting 12-AX in an LSTM network (*Gers99 Krueger 09*)
- Repeated sequential **switch between mappings** (12-AX, AB-X1, XZ-BC,...)
- Non-decreasing switching times
=> **no generalisation** of rules and abstraction of external representations
- Shaping in itself is not sufficient in this architecture
 - results in a different type of abstraction (rules rather than variables)

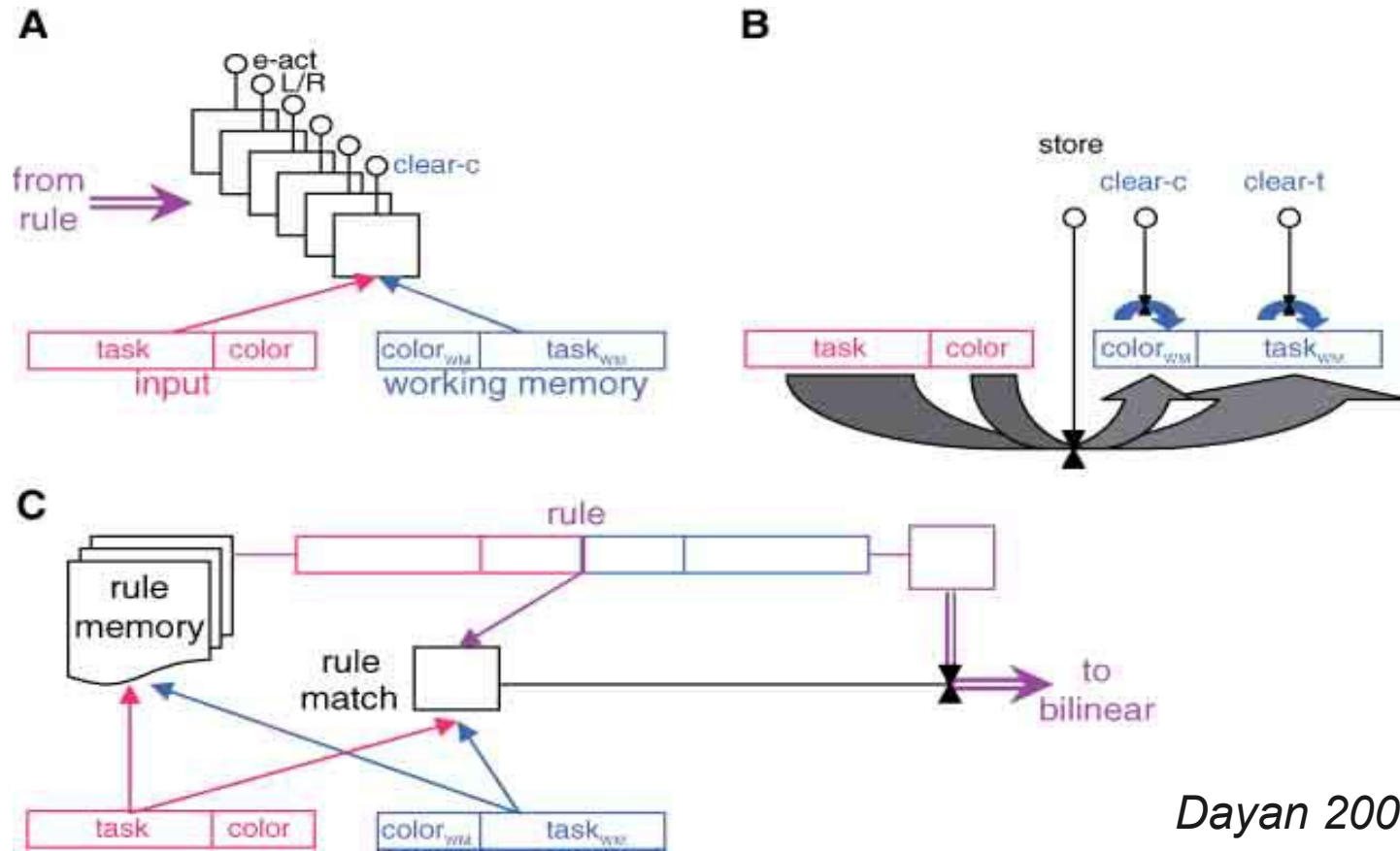
Connectionist symbolic computation

- Ideas of abstract rule based (symbolic) computation with neural like architectures date back to at least the late 80s
- Examples of rule models:
 - BoltzCONS
 - proposals for full production system
 - resembles more the programming language of LISP than a feasible neural implementation
 - A distributed connectionist production system
 - similar in nature: Rules updating working memory and triggered on
 - still quite a complex model
- Instead implement a simple model capturing the ideas of PFC

Rules

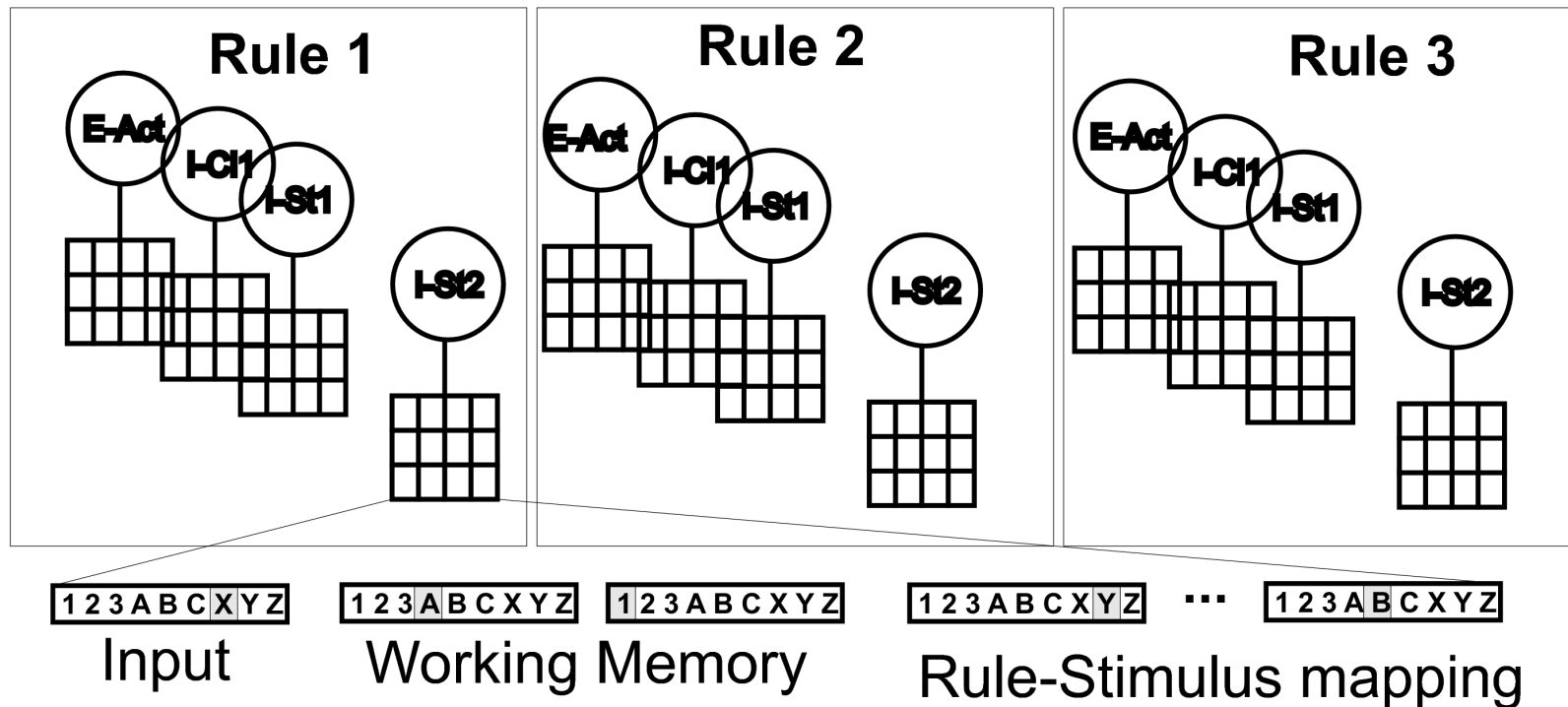
- Divide overall task into a set of simple rules
 - multiple independent rules => disjunction
- Each rule tests for a state condition and executes internal / external actions
 - external actions: observable behaviour
 - internal actions: updating of state (working memory)
- Simple logic-like constructs
 - If *Input = Context-1* Then store *Memory-1*
 - If *Input = PreTarget-1* Then store *Memory-2*
 - If (*Input = Target-1*) and (*Memory-1 = Context-1*) and (*Memory-2 = PreTarget*) Then *Respond-R*
- Define rules in terms of abstract function (Context-1, PreTarget-1), not concrete stimuli (1, A)
- Main operation per rule: (In)Equality, conjunction

A simple model of rule execution



Dayan 2007

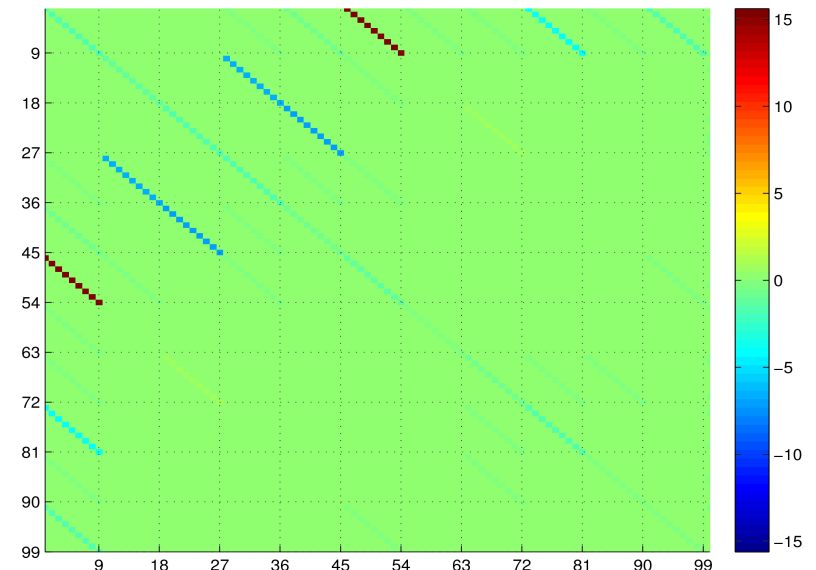
Rule-Stimulus Abstraction



- Stimulus **abstractions** are standard **working memory** slots
- State vector: (current input, 2 working memory slots, 9 rule stimulus mapping slots)
- $P(\text{Act}) = \text{sigmoid}(x W x + w x + b)$
- Each internal / external action has its own weights

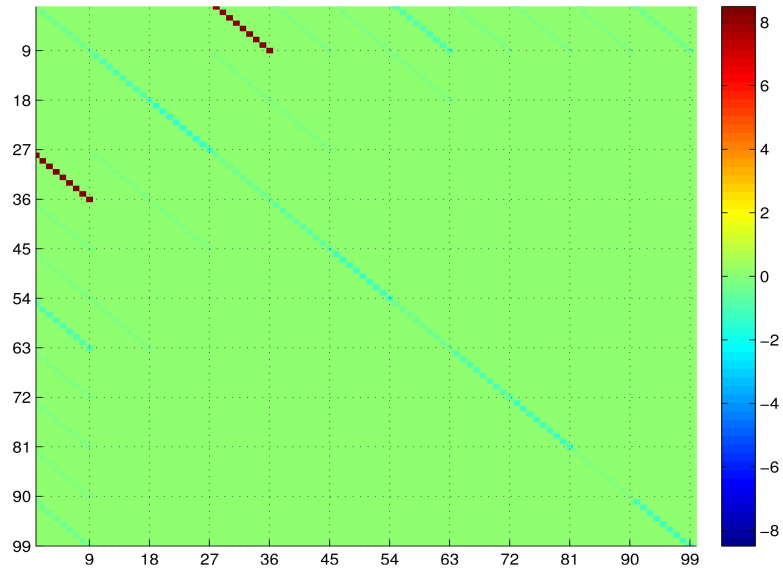
Learning / training

- Supervised, non-sequential training
 - generate set of training examples (e.g. “X | 1 A | 1 A X 2 B Y Z 3” => R)
 - randomly **permute stimulus mappings** and calculate correct response
- Model has a large number of parameters but highly structured and sparse
 - issues with local maxima if trained naïvely
 - apply a l1-regularizer
- Variable stimuli => **No direct input to output dependency**
- Only possible operation: comparison to variable mapping
 - off-diagonal elements can't contribute
- Restrict model to a **multi-diagonal** weight matrix

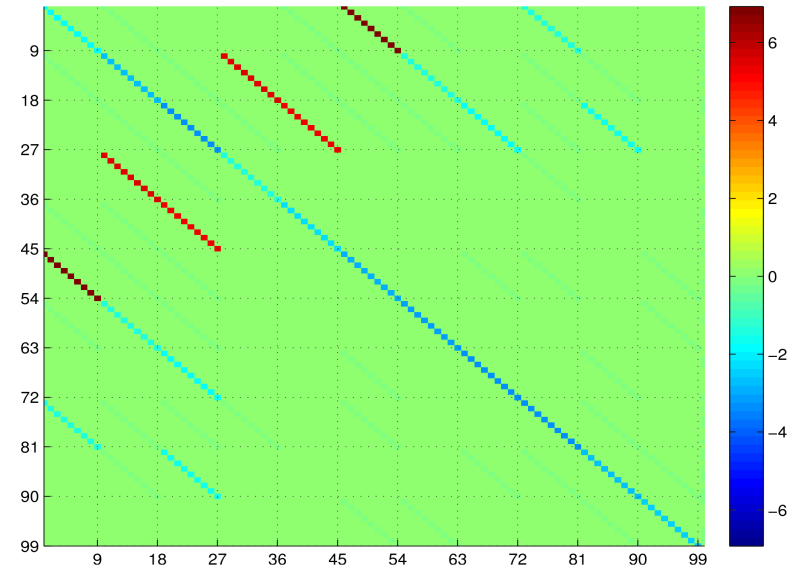


Learned weights

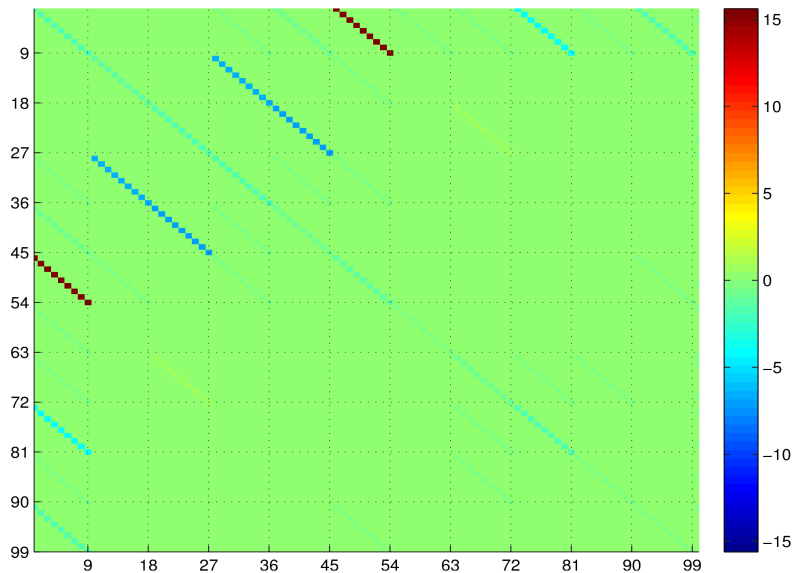
In M1 M2 1 A X 2 B Y C Z



In M1 M2 1 A X 2 B Y C Z



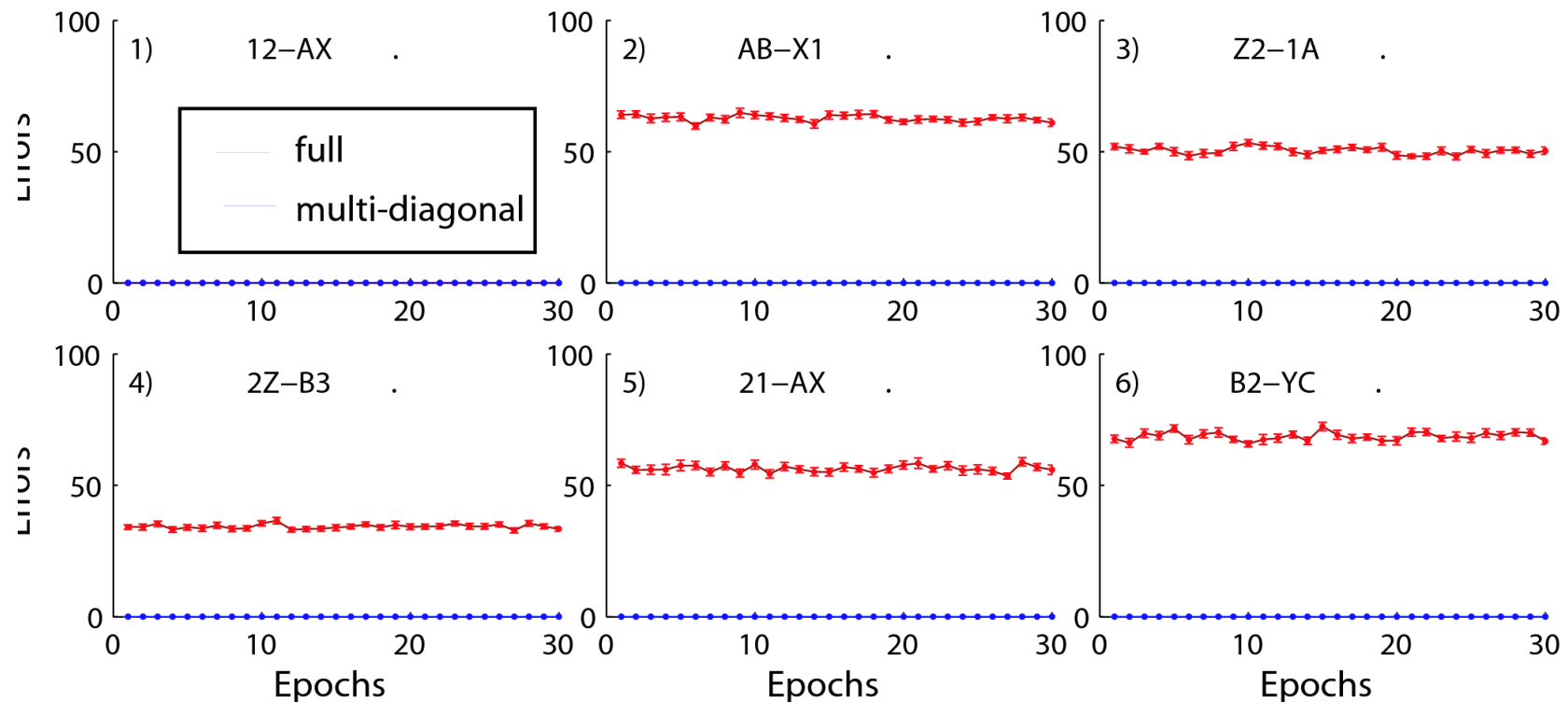
Input = $X \wedge \text{Mem1} = 1 \wedge \text{Mem2} = A$



Input = $X \wedge (\text{Mem1} \neq 1 \vee \text{Mem2} \neq A)$

- Performs task without errors if mappings are loaded correctly
- Reversal as easy as storing new memories
 - $[1 \ A \ X \ 2 \ B \ Y \ C \ Z \ 3] \Rightarrow 12\text{-}AX$
 - $[A \ X \ 1 \ B \ Y \ 2 \ C \ Z \ 3] \Rightarrow AB\text{-}X1$

Automatic generalisation



- **Forced** to generalise by training on a **variety** of stimulus-rule mappings
- Can generalisation occur naturally? I.e. train on one specific instance but still abstract rules from stimuli?
- Requires to **favour matching** against variables over direct inputs
- Proof of concept: Multi-diagonal restriction can achieve this

Habits

- Dayan (2007) modelled habits as a single bilinear form.
 - habitization corresponds to condensing simple individual rules to one combined representation
- Can generalised 12-AX be habitized by this definition?
 - current model is **too limited**
 - can't encode: AX and BY are targets, but AY or BX are not
- Extend model to be more flexible
 - **multi-linear** form => explosion of parameters, tri-linear?, quad-linear?
 - **combinatorial coding**: individual working memory slots represents combinatorial features such as AX
- Debate if all tasks can be habitized or if always need rule like contribution from PFC

Extensions and future work

- Define rules to **update stimulus-rule mappings**
 - incorporate feedback as an additional input
 - more memory required to store a temporal sequence of stimuli
- **Learn** rules in a **sequential** way, equivalent to the task presented to humans
 - requires a form of temporal credit assignment
 - implemented as actor-critic?
 - (self) shaping as a way to learn individual rules
- **Recursive** updating of internal working memory (**Compositionality**)
 - currently modelled as a single **feed-forward** layer per external time step
 - allows more complex tasks while keeping individual rules simple
 - storing non-inputs into working memory
- Interactions between rules and habits

Conclusions

- Abstract rule representations are an important aspect of flexible behaviour, however stimulus abstraction does not naturally arise from traditional weight based learning models without extensive training
- There is a simple solution: Adding a layer of indirection together with explicit representations of working memory. Rules can then act on on stimuli matching working memory rather than on the stimuli directly
- The bilinear framework is one example that is well suited to achieve rule based flexibility.
- Similar ideas likely to apply to other working memory models (PBWM?, LSTM?)
- Can generalize / abstract to new mappings even if initial training was only performed on concrete rules, as long as the abstraction is favoured during learning
- Several open questions remain:
 - what are the implications for sequential learning?
 - what are the computational limits of this model

References

1. Frank M J, Loughry B and O'Reilly R C, **Interactions between the frontal cortex and basal ganglia in working memory: A computational model**. *Cognitive, Affective, and Behavioral Neuroscience*, 1, 2001
2. Dayan P, **Bilinearity, rules, and prefrontal cortex**, *Frontiers in Computational Neuroscience*, 1, 2007
3. Krueger K A and Dayan P, **Flexible shaping: How learning in small steps helps**, *Cognition*, 110, 2009
4. O'Reilly R C and Frank M J, **Making Working Memory Work: A Computational model of learning in Prefrontal Cortex and Basal Ganglia**, *Neural Computation*, 18 (2), 2005
5. Poggio T and Girosi F, **Regularization algorithms for learning that are equivalent to multilayer networks**, *Science*, 1990
6. Rigotti M, Rubin D B D, Wang X-J and Fusi S, **The importance of neural diversity in complex cognitive tasks**, *COSYNE*, 2007
7. Shima K, Isoda M, Mushiake H and Tanji, J, **Categorization of behavioural sequences in the prefrontal cortex**, *Nature*, 445, 2007
8. Touretzky D S, **BoltzCONS: Dynamic symbol structures in a connectionist network**, *Artificial Intelligence*, 46, 1990
9. Touretzky D S and Hinton G E, **A Distributed connectionist production system**, *Cognitive Science*, 12, 1988
10. Wallis J D and Miler E K, **From Rule to response: neuronal processes in the premotor and prefrontal cortex**, *J Neurophysiology*, 2003

Acknowledgments

Support from the Gatsby Charitable Foundation