From AlphaZero to AlphaNPI

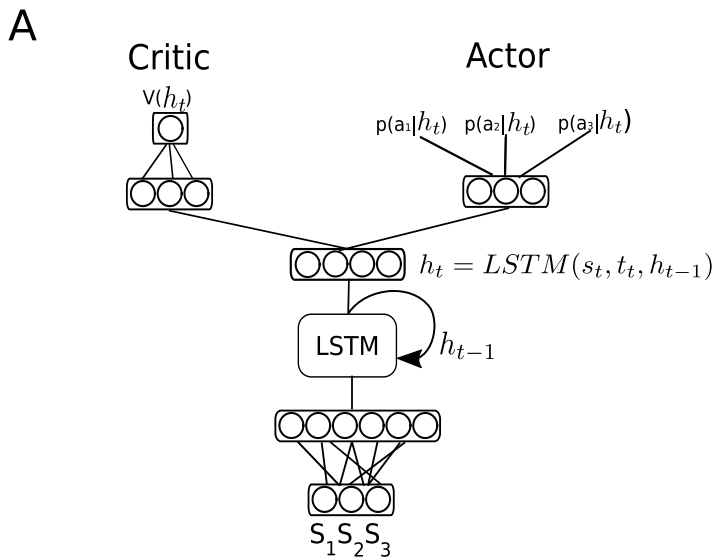
Olivier Sigaud

Sorbonne Université
http://www.isir.upmc.fr/personnel/sigaud

Background

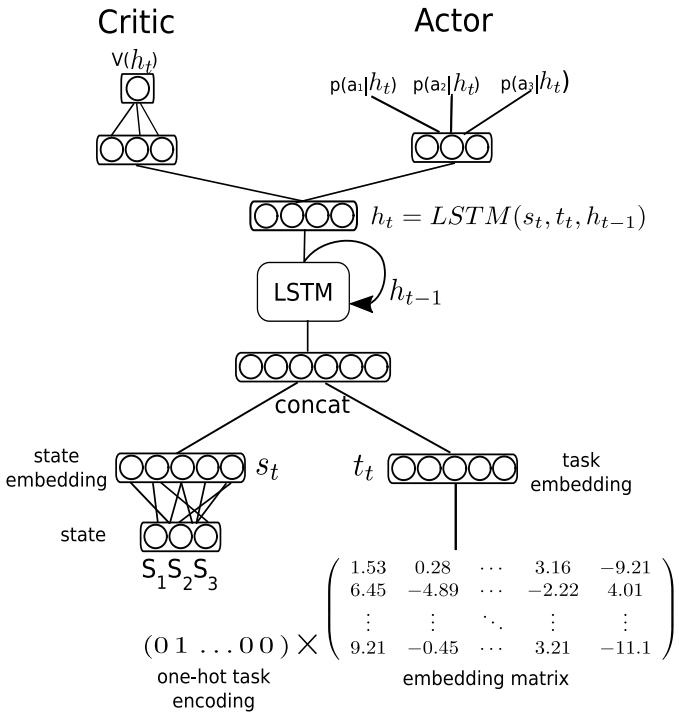
- AlphaZero is very efficient at solving single-task, discrete action problems
- AlphaNPI is an extension to multitask, hierarchical problem solving

Step 1: dealing with non-Markov problems



- An LSTM stores some context from the previous state

Step 2: making it multitask



- Using the task as input makes the architecture multitask (equivalent to GC-RL)
- Additional feature inherited from NPI: using state and task embedding
- Impact of embeddings not studied, ablation needed

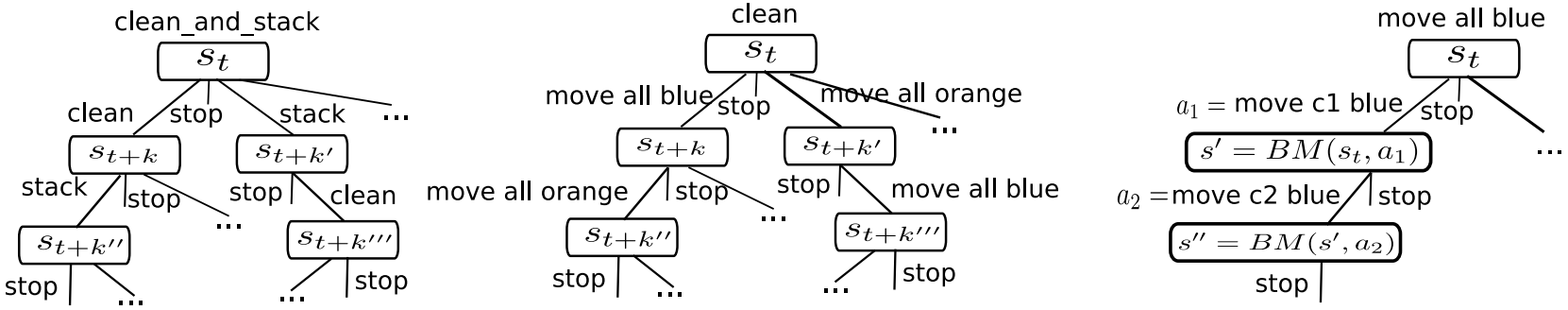
Step 3: defining a (loose) hierarchy of tasks

program	description	level
BUBBLESORT	sort the list	3
RESET	move both pointers to the extreme left of the list	2
Bubble	make one pass through the list	2
RSHIFT	move both pointers once to the right	1
LSHIFT	move both pointers once to the left	1
COMPSWAP	if both pointers are at the same position, move pointer 2 to the left, then swap elements at pointers positions if left element > right element	1
PTR_2_L	move pointer 2 to the left	0
PTR_1_L	move pointer 1 to the left	0
PTR_1_R	move pointer 1 to the right	0
PTR_2_R	move pointer 2 to the right	0
SWAP	swap elements at the pointers positions	0
STOP	terminates current program	0

Table 4: Program library for the list sorting environment.

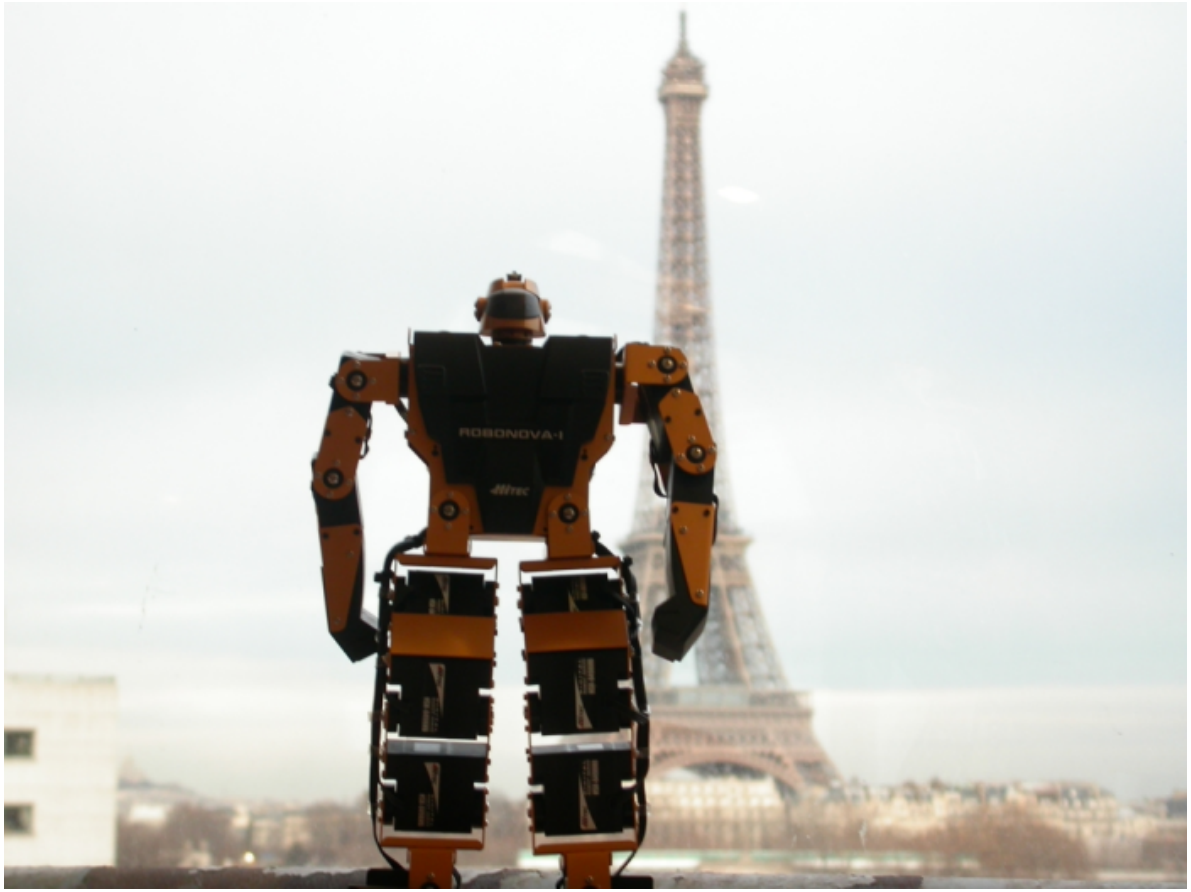
- The list of task is where expert knowledge is inserted
- A task can only call a subtask of lower or equivalent level
- This helps constraining recursive tree search
- Additional constrainsts with preconditions can be used

Recursive tree search: implementation



- When a task calls a subtask
  - A subtree is created
  - The current task context is stored into a stack
  - And unstacked upon termination (as when calling a function in programming languages)
- Thus in AlphaNPI we have a tree of MCTS searches (tree of trees)

Any question?



Send mail to: Olivier.Sigaud@upmc.fr