Layered Learning Techniques for Robot Soccer Movement

Roen Aronson
Division of Science and Mathematics
University of Minnesota, Morris
Computer Science Senior Seminar
arons121@morris.umn.edu

Introduction / Background

- In robot soccer, AI controlled robots compete to score goals on a soccer field.
- Physical tasks are exceptionally difficult for robots and many training techniques are insufficient.
- Layered learning has been used to train these robots to operate more effectively.

Outline

- Robot Soccer
- Layered Learning
- Layered Learning in Robot Soccer Movement
    - Three Layered Learning Strategies
- Learning to dribble: Study by Leottau, et. al.
- Experiment and Results
- Conclusions

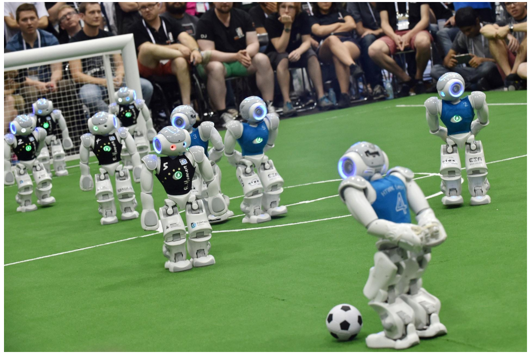
Outline

- **Robot Soccer**
- Layered Learning
- Layered Learning in Robot Soccer Movement
    - Three Layered Learning Strategies
- Learning to dribble: Study by Leottau, et. al.
- Experiment and Results
- Conclusions

Robot Soccer - RoboCup

RoboCup is an organization with the goal of promoting artificial intelligence research. They host tournaments where teams of humanoid robots play soccer against one another. Many advancements in robot control and decision making are due to the influences of RoboCup. They also have leagues for rescue robots.

"By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup."



Robot Soccer - Environment

- Robots are trained in a decentralized environment
- Robots need to be able to communicate and make decisions alone
- Robots need to be able to dribble, pass, and shoot the ball

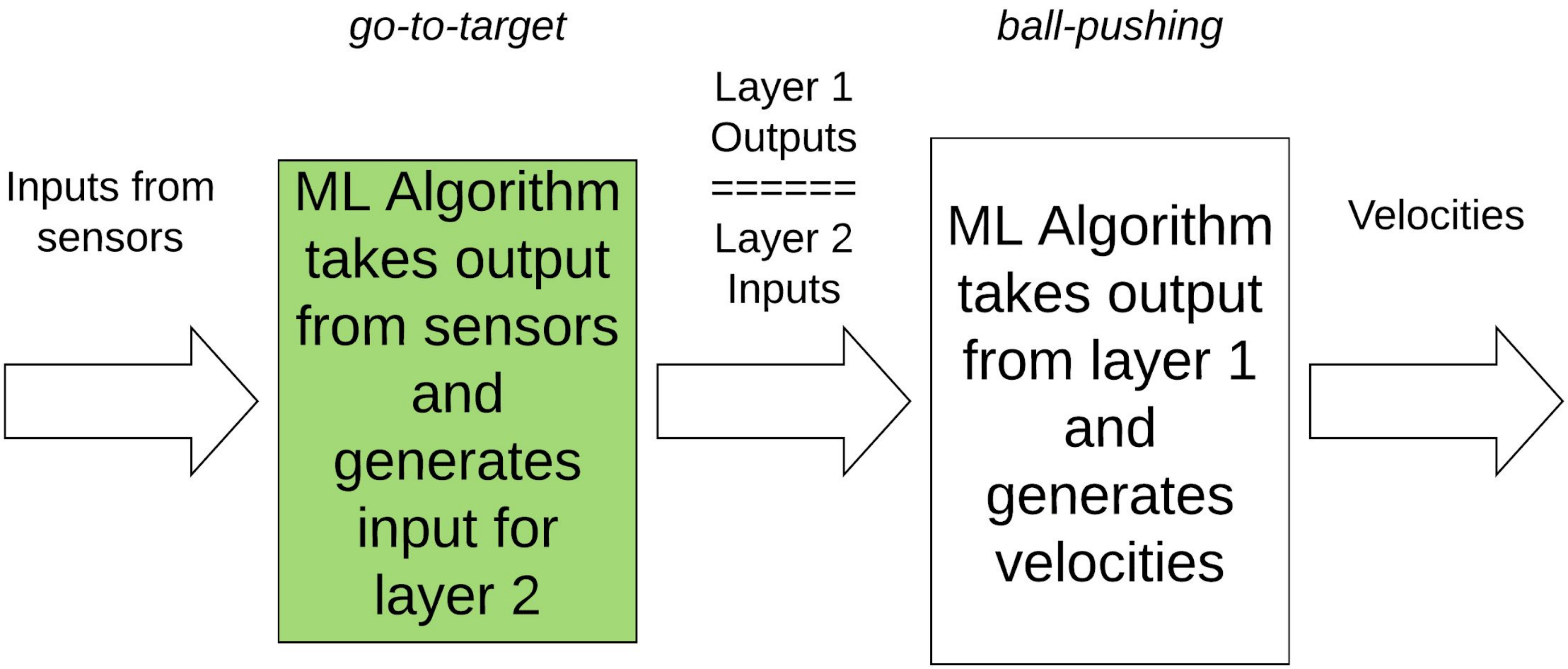
Outline

- Robot Soccer
- **Layered Learning**
- Layered Learning in Robot Soccer Movement
  - Three Layered Learning Strategies
- Learning to dribble: Study by Leottau, et. al.
- Experiment and Results
- Conclusions

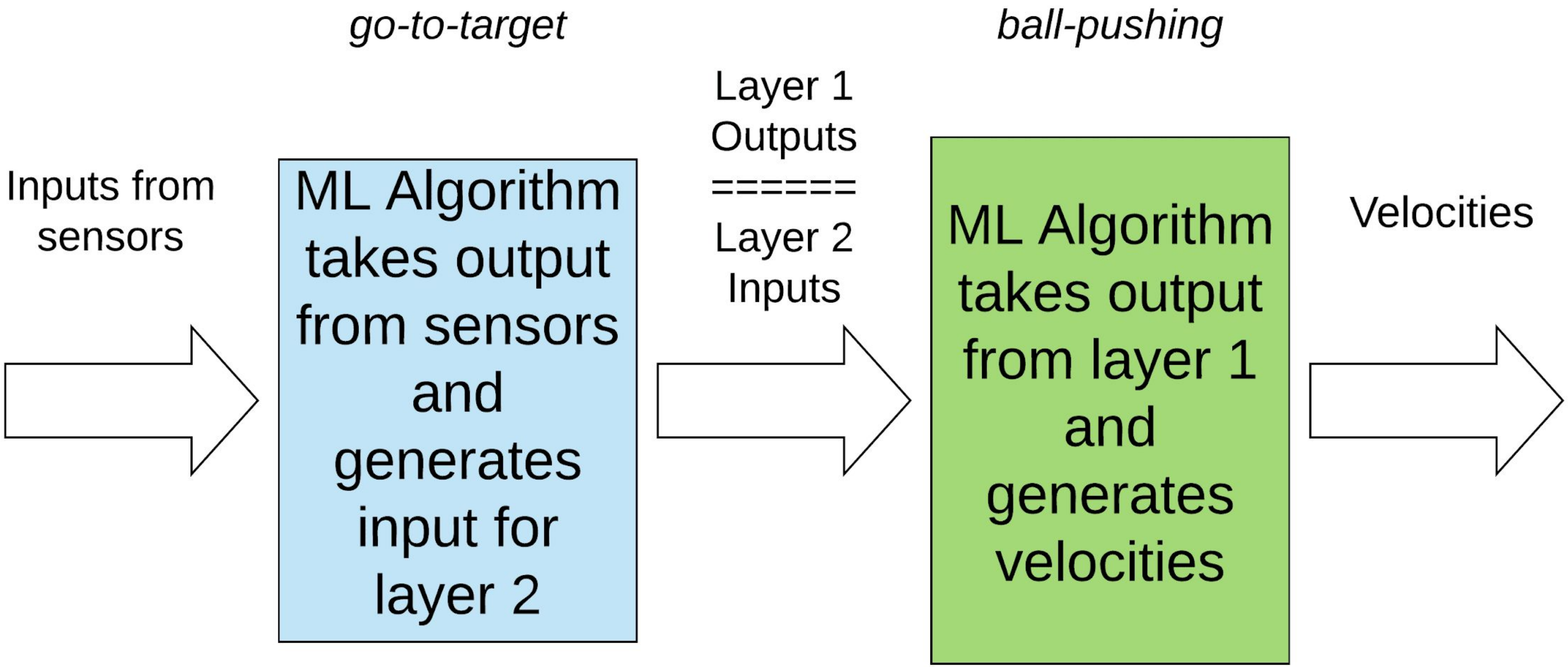
Layered Learning (LL)

- What is layered learning?
  - Behaviors are learned in layers, each layer's depending on the previous layer's learning
  - In each layer, a different machine learning approach is hand-picked for the task
  - Learn complex tasks by breaking it down into smaller sub-tasks
- Who is using layered learning?
  - RoboCup
  - Image Processing
  - Speech Recognition

First Layer Learning



Second Layer Learning



Machine Learning (ML) Background

- ML is the idea that algorithms can improve through experience.
- Reinforcement learning (RL) is a subset of machine learning where reward functions are used in order to incentivize “good” behavior.
- Essentially, a ML algorithm should find a way to map inputs to outputs without explicit programming.

Continued on next slide.

Machine Learning (ML) Background

- ML algorithms work by searching a state space for solutions.
- A state space (search space) is a representation of every possible state that the algorithm can search.
- When variables are not allowed to influence the outcome of an algorithm, they are considered to be outside of the state space.

Picking ML Algorithms

- Each layer uses a different, hand-picked algorithm to train a task.
- An algorithm could be anything from the very simple to advanced neural networks.
- Algorithms are picked that suit the problem. This helps tasks learn faster because they can be broken down and the most effective algorithm can be used to solve the sub-task.

Outline

- Robot Soccer
- Layered Learning
- **Layered Learning in Robot Soccer Movement**
  - Three Layered Learning Strategies
- Learning to dribble: Study by Leottau, et. al.
- Experiment and Results
- Conclusions

Layered Learning in Robot Soccer Movement

- Researchers use layered learning to teach robots to dribble
- Why is layered learning used?
    - Dribbling is complex and not easily learned.
    - Mechanics of movement are not necessarily understood.
- Who is using layered learning?
    - Many RoboCup teams around the world, including Universities such as UChile. The idea was popularized in the early 90's.
- What tasks are solved with layered learning?
    - Dribbling, Passing, Getting up, Shooting, Kick-off

Why is Layered Learning Used?

- Problem is not understood by humans
  - Simple physical actions are actually exceptionally difficult to perform by robots
- Machine learning has been proven in the field of AI
  - Methods similar to it have been used in other fields of AI, so there was already promise
- Problem can be broken down into simpler sub-problems, which lends itself to LL

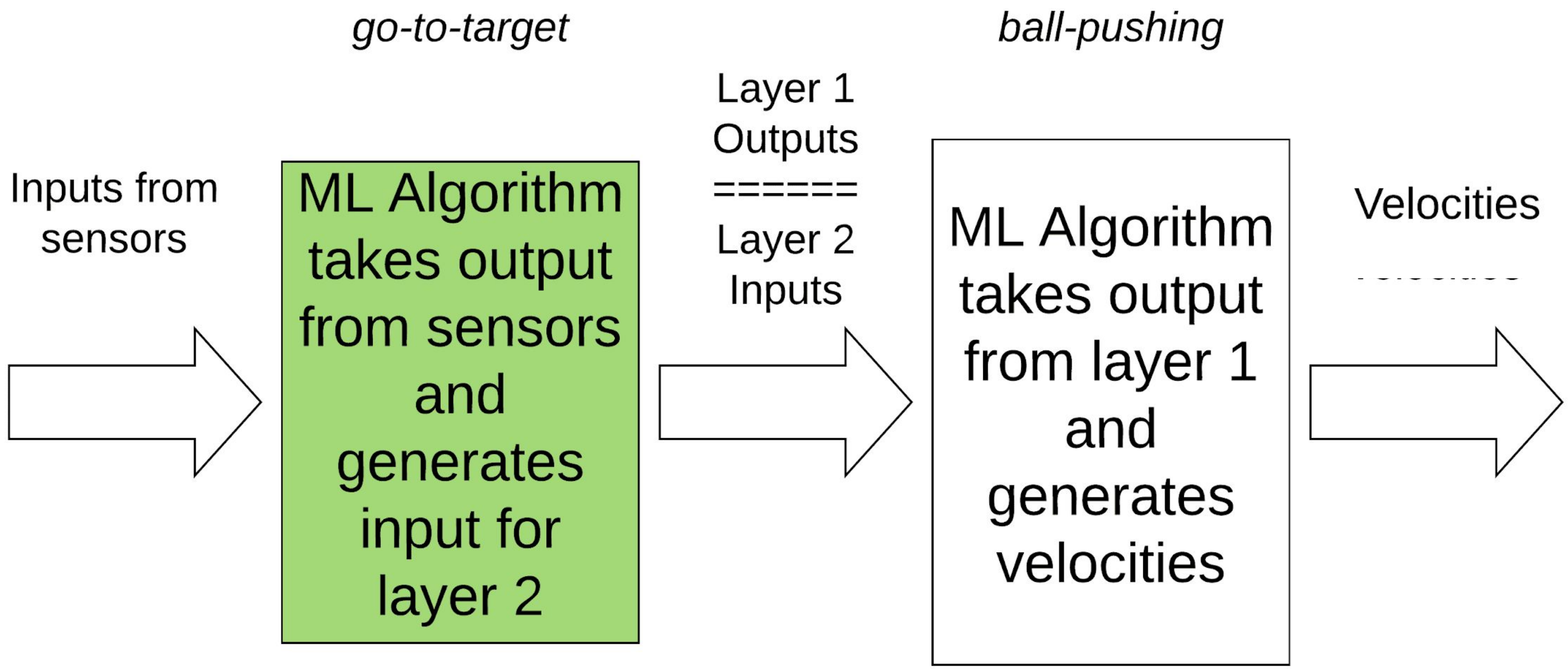
Outline

- Robot Soccer
- Layered Learning
- Layered Learning in Robot Soccer Movement
    - **Three Layered Learning Strategies**
- Learning to dribble: Study by Leottau, et. al.
- Experiment and Results
- Conclusions

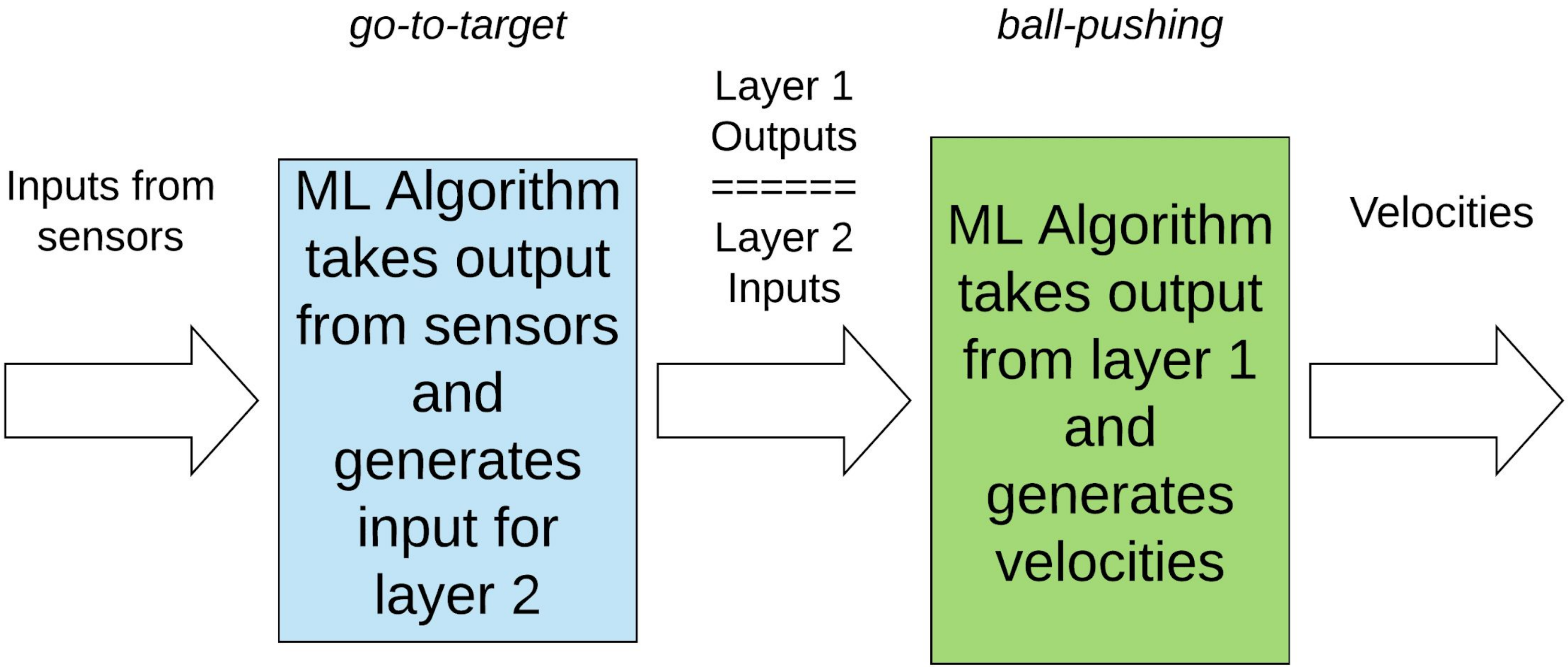
Sequential Layered Learning (SLL)

- Layers learned in order from left to right
- Previous layers are frozen (halted) while learning later ones
- Earliest iteration of LL

First Layer Learning



Second Layer Learning



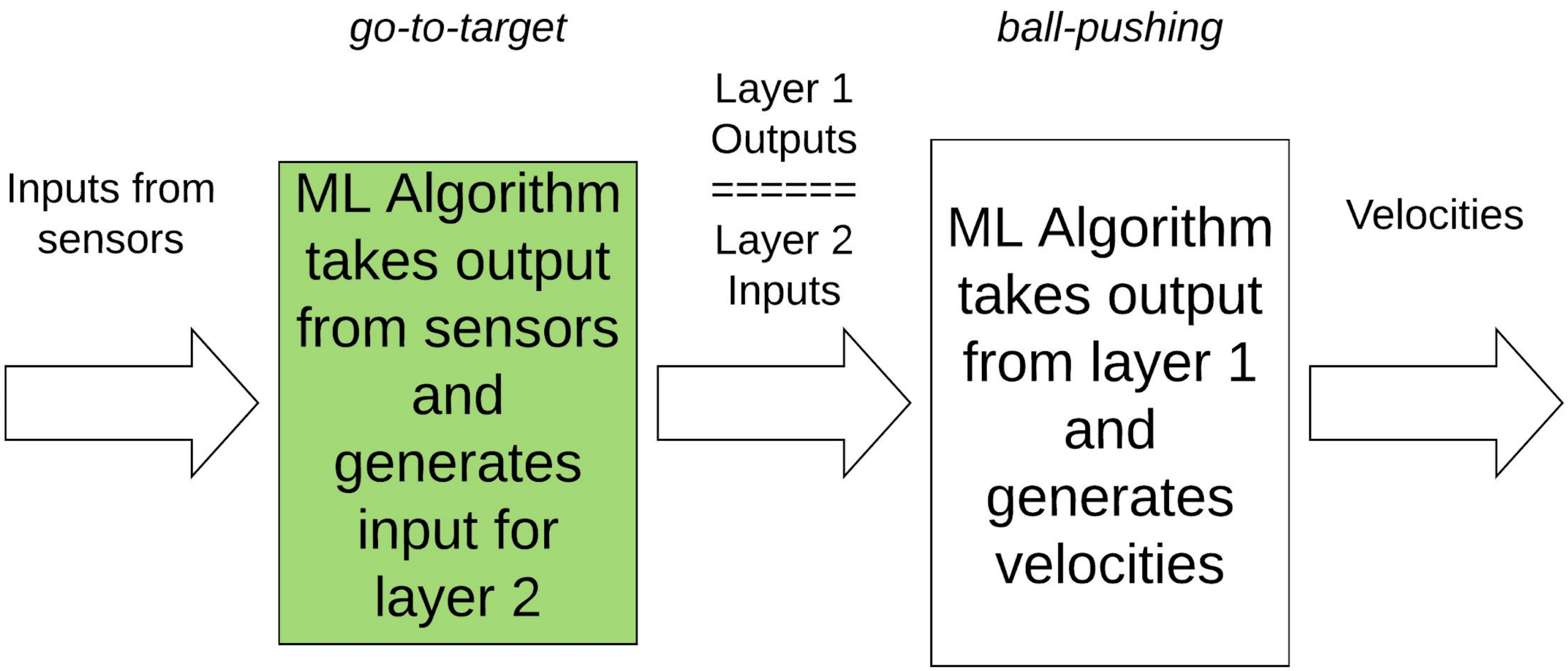
Sequential Layered Learning (SLL)

- Pros
  - Reduces the search space
  - Searching can be faster
- Cons
  - Limits number of possible solutions

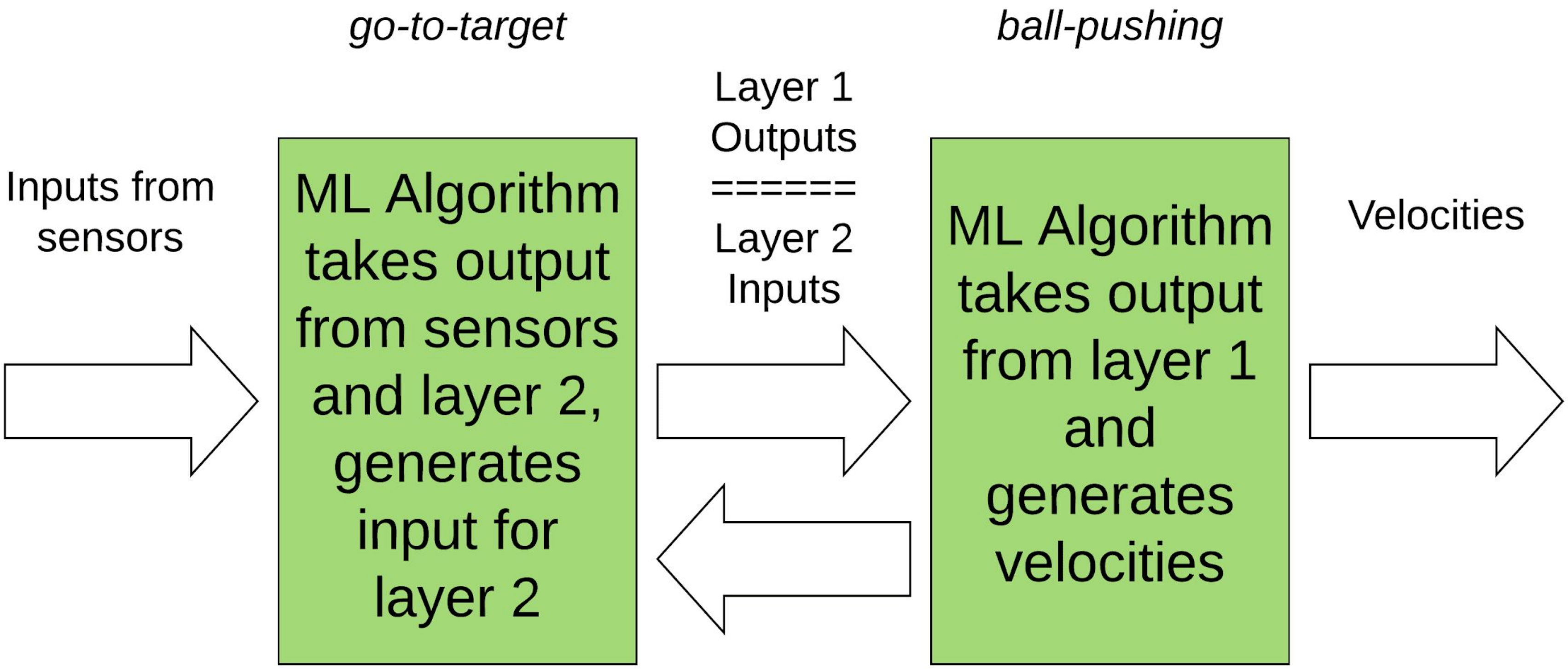
Concurrent Layered Learning (CLL)

- Previous layers continue to learn while later layers are being trained
- May explore layer's joint space in order to optimize behaviors learned in previous layers.

First Layer Learning



Second Layer Learning



Concurrent Layered Learning (CLL)

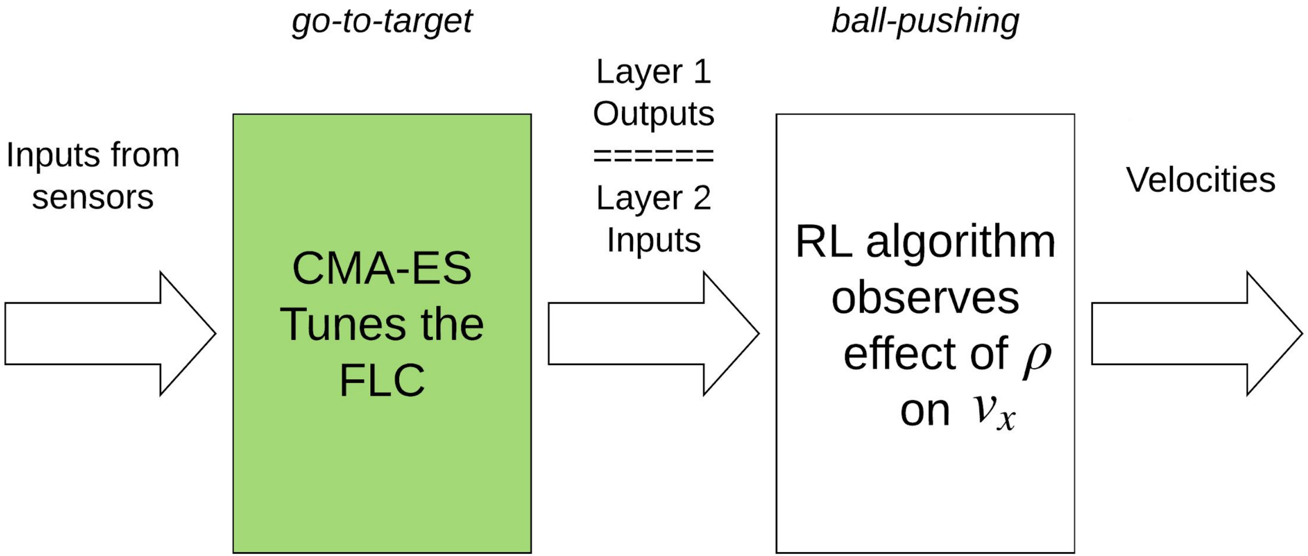
- Pros
  - Search larger space
  - More possible solutions
- Cons
  - Slower learning
  - More difficult learning (correct solutions may be harder to find)

Partial Concurrent Layered Learning (PCLL)

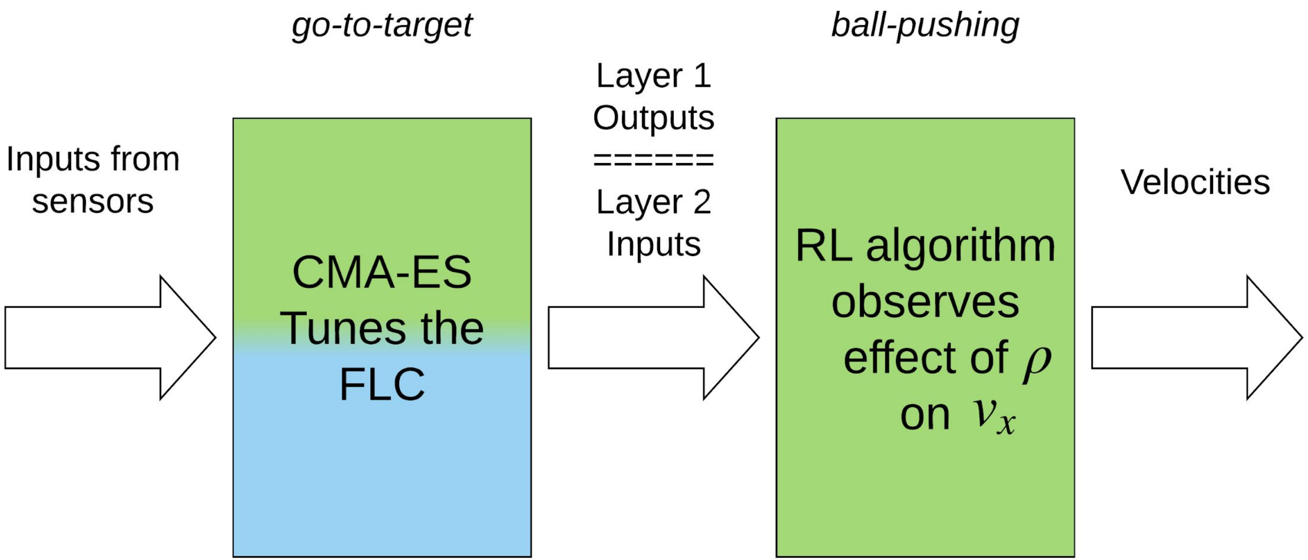
- Middle ground between SLL and CLL strategies
- Freezes parts of previous layers, allowing other parts to continue learning
- Parts from previous layers “overlap” with the next layer

- Attempts to alleviate cons of both previous strategies

PCLL First Layer Learning



PCLL Second Layer Learning



Outline

- Robot Soccer
- Layered Learning
- Layered Learning in Robot Soccer Movement
- **Learning to dribble: Study by Leottau, et. al.**
- Experiment and Results
- Conclusions

Leottau et. al.

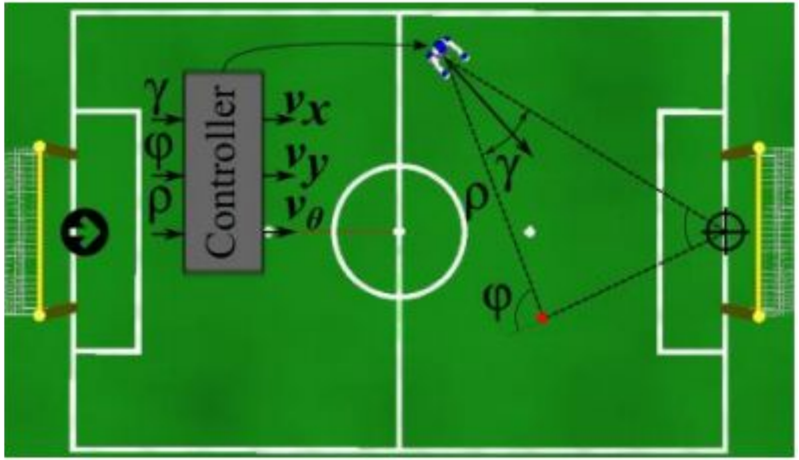
- Ball dribbling is learned in layers
- Three different approaches to LL strategies that all use the same control scheme
  - Partial Concurrent Layered Learning (specific form of OLL) [2]
  - SLL
  - CLL
- Results are compared and the best and worst parts of each approach are explained

Leottau et. al. - Control Actions

Control Actions:

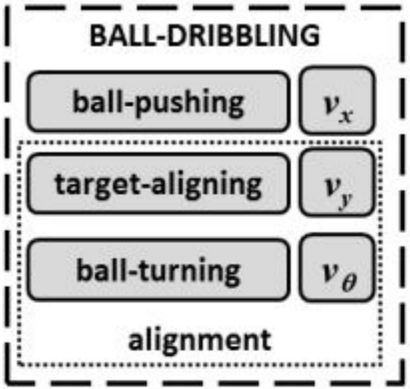
- ρ , γ , φ - robot-ball distance, robot-ball angle, and alignment angle

- $[v_x\,, v_y\,, v_\theta]'$ - robot linear and angular velocities



Leottau et. al. - Behaviors Learned

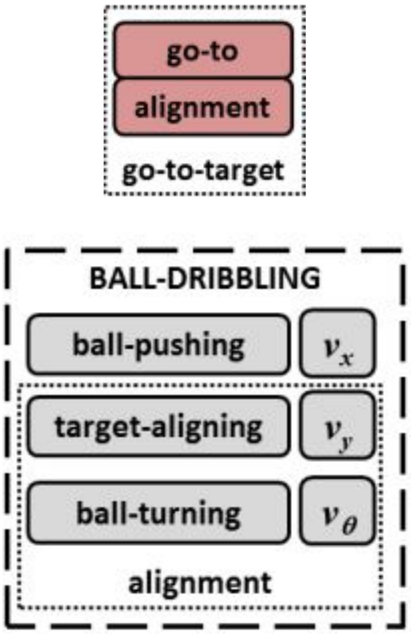
- The BALL-DRIBBLING task is separated into multiple sub-tasks:
  - *ball-pushing* - v_x, the robot's forward speed towards the ball
  - *alignment* - combination of two behaviors:
    - *target-aligning* - v_y, the speed at which the robot moves laterally
    - *ball-turning* - v_θ, the speed at which the robot turns



(a)

Figure (a) shows how different behaviors relate to each-other. Note that dribbling is essentially ball-pushing combined with alignment.

Leottau et. al. - Behaviors Learned



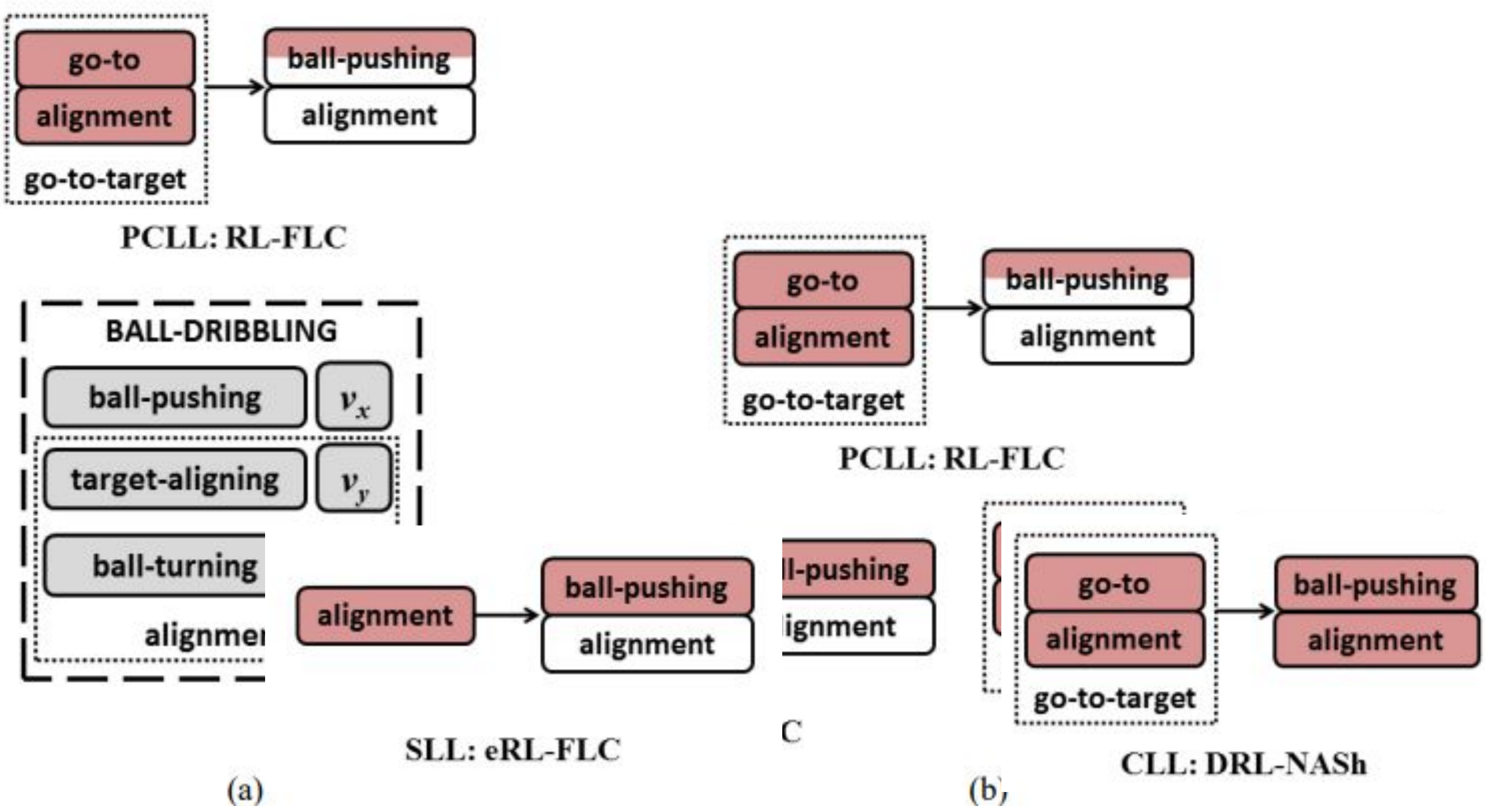
- *go-to-target* is a similar behavior to *ball-dribbling*, but instead of using ball-pushing as it's movement forward, it uses *go-to*.
    - *go-to* - v_x , the robots forward velocity towards a target
- Ball-dribbling is the forward-moving behavior that involves moving the ball, and go-to-target is the forward-moving behavior that does not involve moving the ball.

Figure (a) shows how different behaviors relate to each-other. Note that dribbling is essentially ball-pushing combined with alignment.

(a)

Leottau et. al. - Behaviors Learned cont.

- Prior to each of the three LL strategies being applied to the dribbling problem, the *go-to* behavior is learned via a fuzzy logic controller (FLC), which is tuned using an algorithm called the Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
- CMA-ES
    - An evolutionary computation strategy that recombines code in order to optimize the solution to a problem.
- FLC
    - Takes inputs from the sensors, and uses CMA-ES in order to find accurate parameters in order to generate outputs.

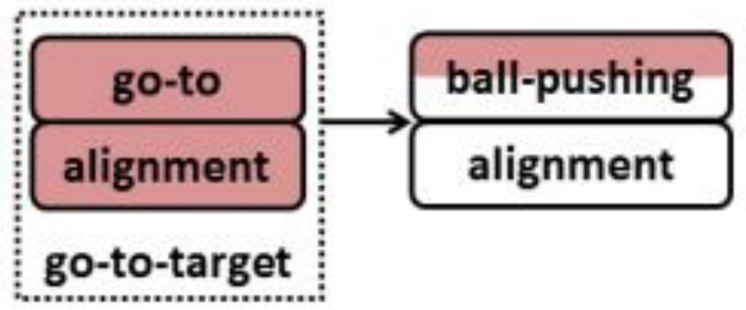


go-to
alignment
go-to-target
ball-pushing
alignment
PCLL: RL-FLC
BALL-DRIBBLING
ball-pushing
v_x
target-aligning
v_y
ball-turning
alignmen
alignment
ball-pushing
alignment
SLL: eRL-FLC
(a)
go-to
alignment
go-to-target
ball-pushing
alignment
PCLL: RL-FLC
ll-pushing
ignment
C
go-to
alignment
go-to-target
ball-pushing
alignment
CLL: DRL-NASh
(b)

Behavior	LL Strategy	Learned in Layer 1	Learned in Layer 2
go-to	- - PCLL	FLC parameters of v_x	-
alignment *go-to-target* *Dribbling with RL-FLC*	- - PCLL	FLC parameters of v_y and v_θ *go-to* and *alignment* *go-to-target*	- - Partial policy for v_x observing ρ RL algorithm
Dribbling with eRL-FLC	SLL		
Dribbling with DRL-NASh	CLL		

Leottau et. al. - Partial Concurrent LL (PCLL)

- In the first layer, a Fuzzy Logic Controller (FLC) is tuned by training *go-to-target*.
- In the second layer, a partial policy for v_x is learned.
    - Alignment is completely frozen
    - Only the parameter for how ρ affects v_x is left open, while γ and φ are frozen.
    - This means that ball-pushing is learned in the context that alignment cannot change, when normally ball-pushing is a behavior that relies on alignment



PCLL: RL-FLC

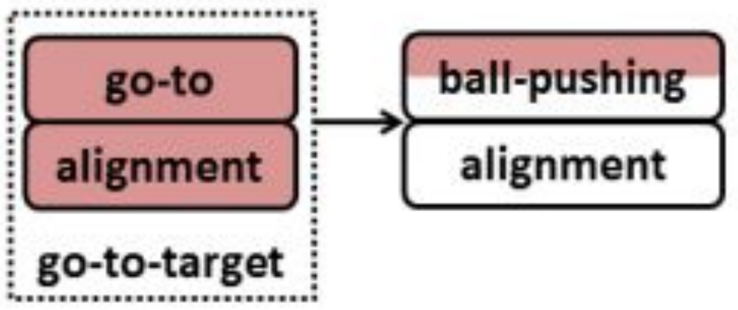
Continued on next slide

Leottau et. al. - Partial Concurrent LL (PCLL)

- The proposed reward function when learning this behavior is simple:

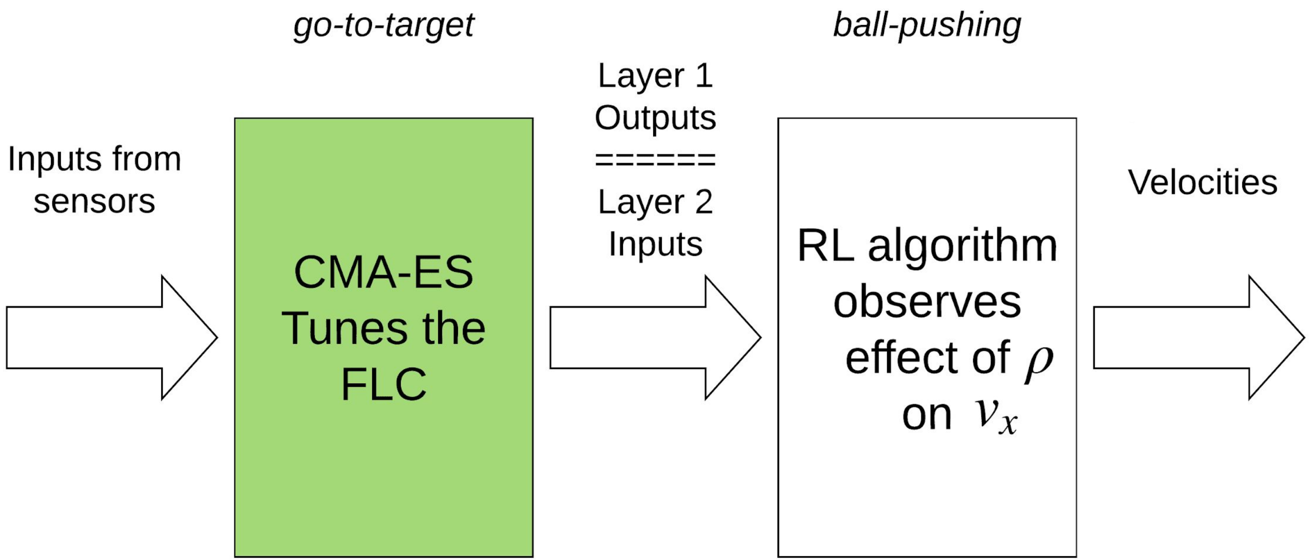
$$r_x = \begin{cases} 1, & \rho < \rho_{th} \wedge v_x \geq v_{x.max}, \\ -1, & otherwise \end{cases}$$

- This means the robot keeps moving forward at a reasonable speed (150 mm/s), and the ball never gets too far away from it (500 mm).

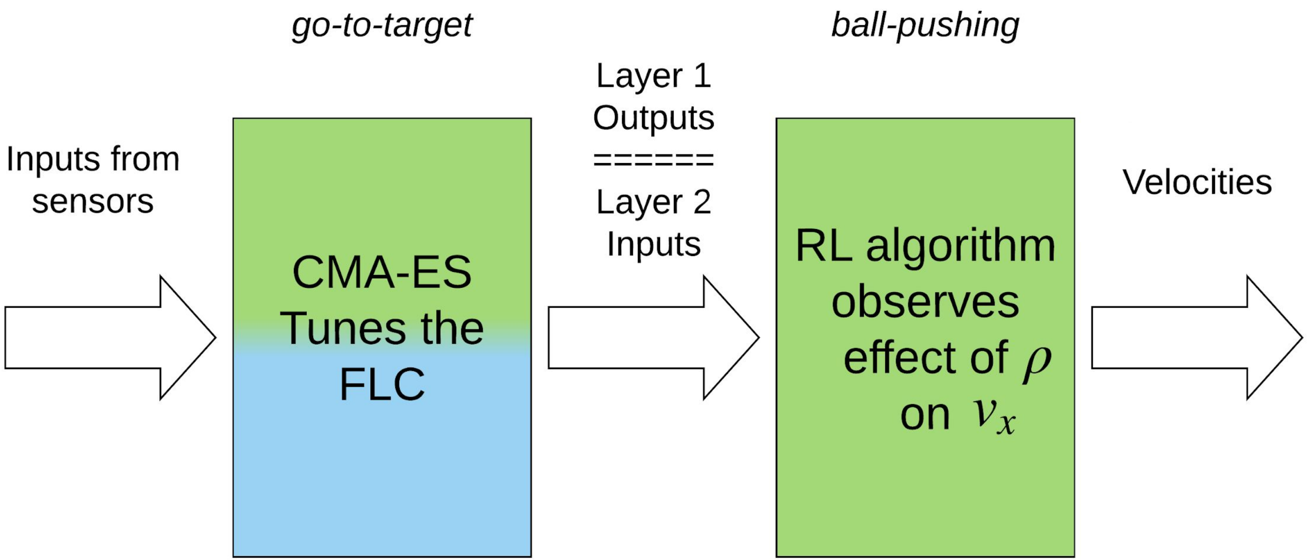


PCLL: RL-FLC

PCLL First Layer Learning

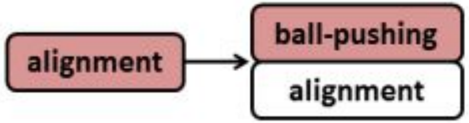


PCLL Second Layer Learning



Leottau et. al. - Sequential LL (SLL)

- In the first layer, alignment is learned in a similar fashion to the PCLL strategy.
- In the second, an enhanced RL technique is used in order to include the entire state space [ρ,γ , φ].
- This is done in order to include scenarios in which the alignment angle, γ , is not 0. In the PCLL strategy, the ideal scenario is assumed.



SLL: eRL-FLC

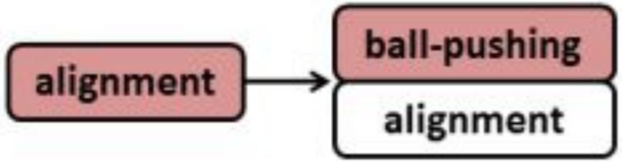
Continued on next slide

Leottau et. al. - Sequential LL (SLL)

- The proposed reward function is as such:

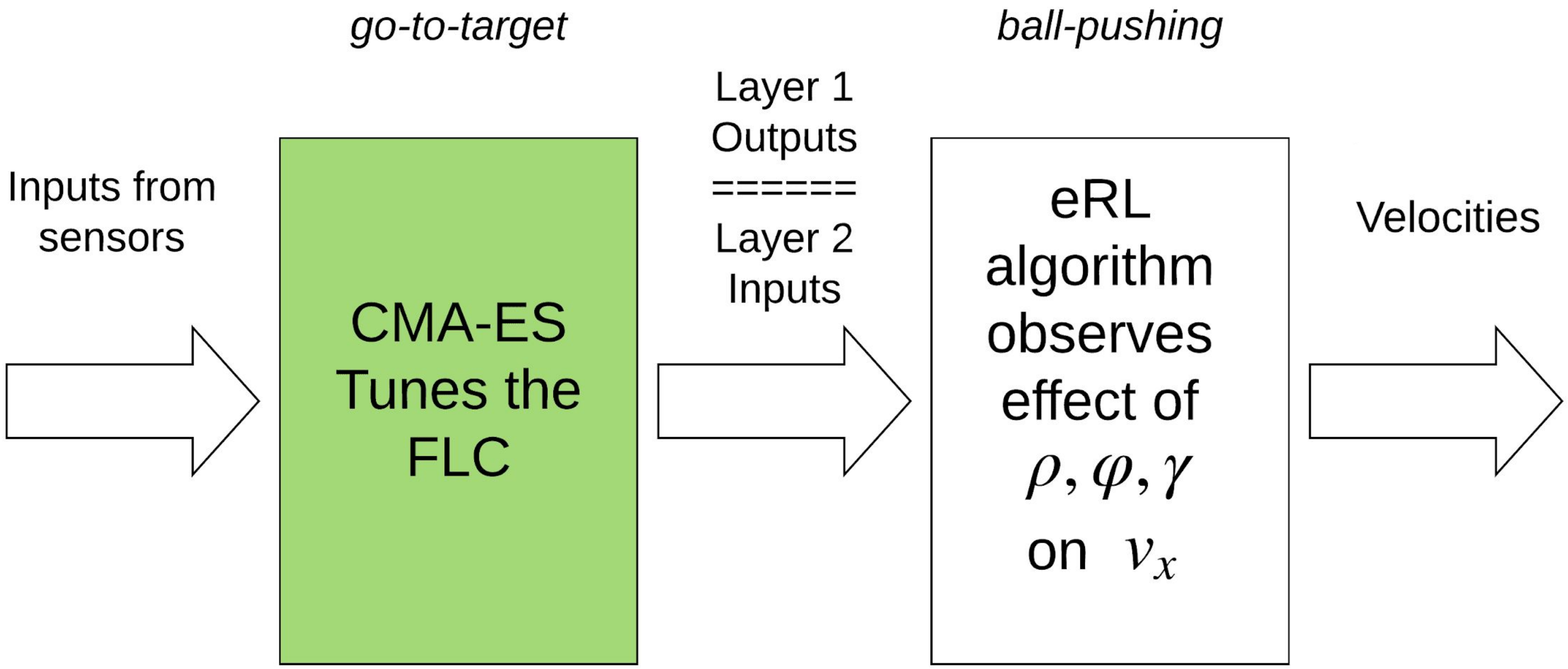
$$r_x = \begin{cases} 1, \rho < \rho_{th} \wedge |\gamma| < \gamma_{th} \wedge |\varphi| < \varphi_{th} \wedge v_x \geq v_{x.max}, \\ -1, \qquad\qquad\qquad\qquad\qquad\qquad otherwise \end{cases}$$

- This means that not only do the restrictions from PCLL apply, but the robot and ball must also stay within -15° and 15° of the target line.

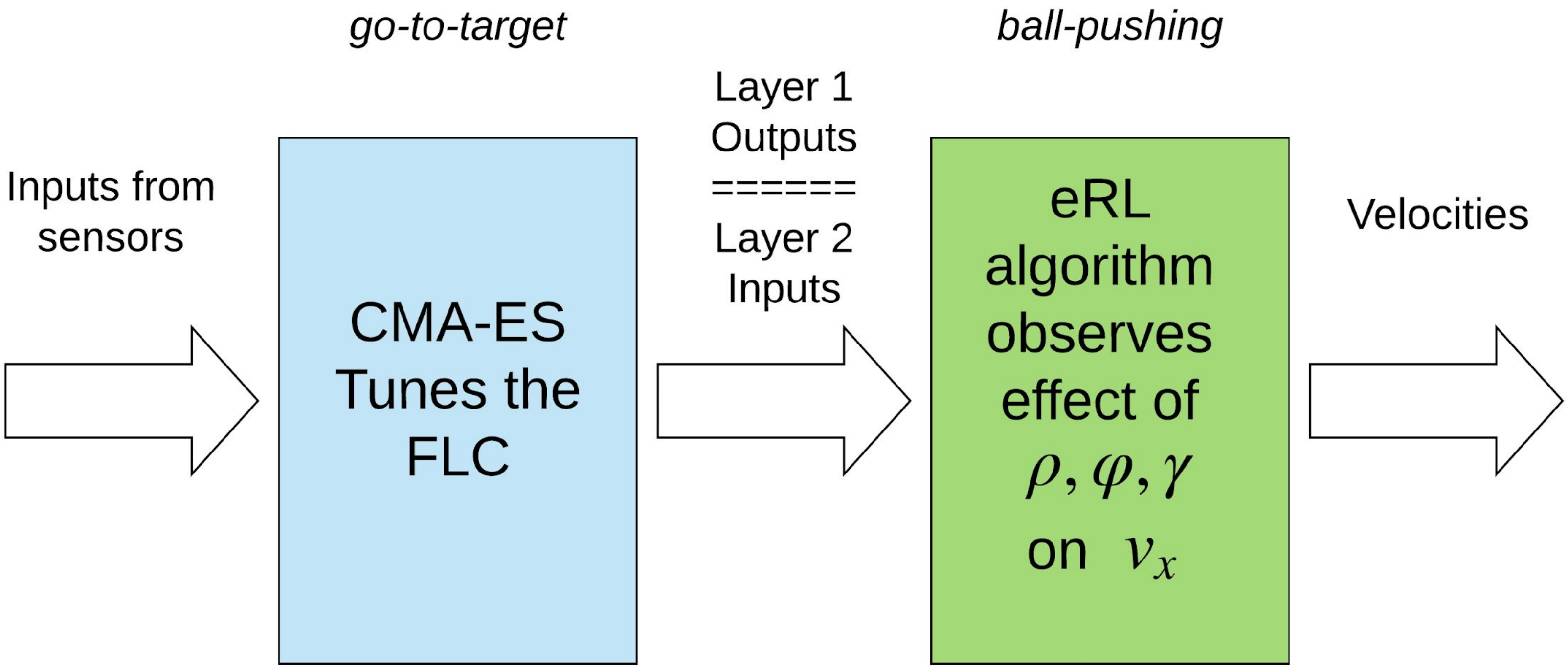


SLL: eRL-FLC

SLL First Layer Learning

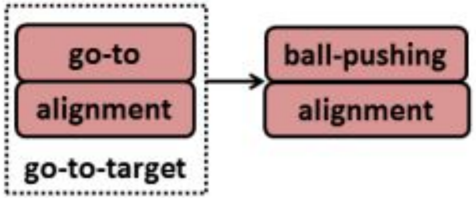


SLL Second Layer Learning



Leottau et. al. - Concurrent LL (SLL)

- In the first layer, *go-to-target* is learned using a decentralized reinforcement learning technique.
- In the second layer, a Nearby Action Sharing (NASh) approach is used in order to train *ball-pushing* and *alignment*.
  - Three components, v_x, v_y, v_θ, learn policies in parallel.
- Policies are trained on these shared parameters:
  - 0mm> ρ > 500mm
  - -15°> γ >15°
  - -15°> φ >15°

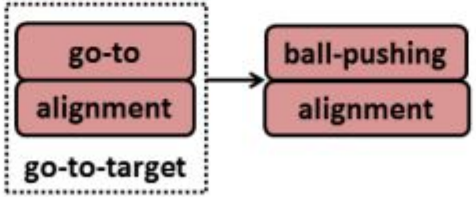


CLL: DRL-NASh

Continued on next slide

Leottau et. al. - Concurrent LL (SLL)

- The NASh approach allows for knowledge from previous layers to be applied to current layer's learning, while at the same time searching for new knowledge in the search space.
- Each policy is able to learn by observing the joint state space : [ρ, γ, φ]



CLL: DRL-NASh

Continued on next slide

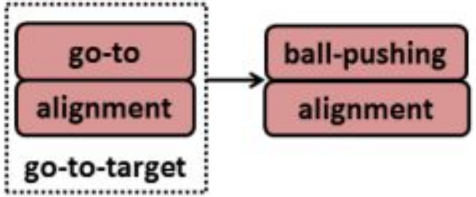
Leottau et. al. - Concurrent LL (SLL)

The reward functions for each agent are as such:

$$r_x = \begin{cases} 1, \rho < \rho_{th} \wedge |\gamma| < \gamma_{th} \wedge \ |\varphi| < \varphi_{th} \wedge v_x \geq v_{x.max'} \\ -1, \qquad \qquad \qquad \qquad \qquad \qquad \qquad otherwise \end{cases}$$

$$r_y = \begin{cases} 1, \ |\gamma| < Ang_{th} \\ -1, \qquad otherwise \end{cases}$$

$$r_\theta = \begin{cases} 1, \ |\gamma| < Ang_{th} \wedge |\varphi| < Ang_{th} \\ -1, \qquad \qquad \qquad \qquad otherwise \end{cases}$$



CLL: DRL-NASh

Outline

- Robot Soccer
- Layered Learning
- Layered Learning in Robot Soccer Movement
- Learning to dribble: Study by Leottau, et. al.
- **Experiment and Results**
- Conclusions

Leottau et. al. - Experiment and Results

- Four different dribbling schemes are compared in simulation:
  - Partial Concurrent Layered Learning
  - Sequential Layered Learning
  - Concurrent Layered Learning
  - A RL technique that does not involve layered learning, as a basis of comparison
- The evolution of the learning processes are evaluated by averaging 10 runs each, by three global parameters:
  - Percentage of Forward Maximum Speed, $\%S_{F_{max}}$
  - Percentage of Time in a Faulty State, $\%T_{FS}$
  - Global Fitness, $1/2 * [(100 - \%S_{F_{max}}) + \%T_{FS}]$ = F, where F=0 is the desired policy.

Leottau et. al. - Experiment and Results

- Each experiment is conducted with these parameters as fault states:
  - $[\rho_{th}, \gamma_{th}, \varphi_{th}] = [500mm, 15°, 15°]$
  - $\text{Ang}_{th}=5°$
  - $v_{x.max'} = 0.9 \cdot v_{x.max}$,

Leottau et. al. - Experiment and Results

Method	$\%S_{Fmax}$		$\%T_{FS}$		F	*Time to Th. (Episodes)*
	Avg.	*Std.Err*	*Avg.*	*Std.Err*	*Avg.*	
DRL-NASh (CLL)	74.83	0.049	14.69	0.080	19.92	1391
eRL-FLC (SLL)	61.49	0.032	16.84	0.061	27.67	66
RL-FLC (PCLL)	57.50	0.04	26.32	0.069	34.4	53
DRL	64.35	0.12	13.87	0.19	24.76	1594

Outline

- Robot Soccer
- Layered Learning
- Layered Learning in Robot Soccer Movement
- Learning to dribble: Study by Leottau, et. al.
- Experiment and Results
- **Conclusions**

Leottau et. al. - Conclusions

- The CLL model reached the best fitness by far, with a fitness score of 19.92, but spent an average of 1391 episodes to reach that.
- The PCLL model reached the desired fitness score with only 53 runs, but failed to perform well over long experiments. This is hypothesized to be a result of freezing parts of the first layer while learning the second layer, and could possibly be improved by freezing other parts or combining with other learning strategies in the first layer.
- The SLL strategy allows for extremely accurate dribbling, but it is also slow. This is because it learns ball-dribbling with respect to alignment.

Acknowledgements

Nic Mcphee - Advisor

KK Lamberty - Senior Seminar Professor

Questions?

Citations

1. **Leottau, David L., et al. "A study of layered learning strategies applied to individual behaviors in robot soccer." *Robot Soccer World Cup*. Springer, Cham, 2015.**
2. Patrick MacAlpine, Mike Depinet, and Peter Stone. **UT Austin Villa 2014: RoboCup 3D Simulation League Champion via Overlapping Layered Learning**. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2842–48, January 2015.