Reinforcement Learning for Intelligent Control
Part 2

Presented at
Chinese Youth Automation Conference
National Academy of Science, Beijing, 8/22/05

George G. Lendaris
NW Computational Intelligence Laboratory
Portland State University, Portland, OR
“Intelligent” aspect of Control:

Why, after 50+ years of doing AI don’t we have autonomous “intelligent” robots walking around, doing useful tasks for us?

A key issue of Machine Learning is called the **Frame Problem**.

Propose Reinforcement Learning to help solve this (fundamental) problem
SO FAR: The RL/ADP has focused on creating optimal policies for specific problems (cf Part 1).

In AI parlance this is like creating a frame or a schema
...and, unfortunately, means we must face the Frame Problem
...but, fortunately, we posit that this Problem can be solved using RL/ADP approach
\[ \forall s \forall x (\text{BLOCK}(x)[s] \supset \text{BLOCK}(x)[\text{GRASP}[s]]) \]

...we were obliged to add the hypothesis that if a person has a telephone he still has it after looking up a number in the telephone book.

(McCarthy & Hayes, 1969)
As knowledge increases so does the computational cost. As a result efficiency decreases.

But….  
As humans gain more knowledge they become more efficient
In the confined world of a robot, surroundings are not static. Many varying forces or actions can cause changes or modifications to it. The problem of forcing a robot to adapt to these changes is the basis of the frame problem in artificial intelligence.

Information in the knowledge base and the robot's conclusions combine to form the input for what the robot's subsequent action should be. A good selection from its facts can be made by discarding or ignoring irrelevant facts and ridding of results that could have negative side effects. (Dennet 1987)
INPUT → FRAME → ACTION

INPUT → POLICY → ACTION
Using neural networks to implement policies gives us the ability to address policies by their weights. Consider a network with parameters $w$.

$$o = N(w, i)$$

Any two set of weights $w = w_1$ and $w = w_2$ identify two unique networks and thus also identify, in general, two unique policies.
Suppose we split the parameters $\mathbf{w}$ into two sets. The first set $\mathbf{w}_m$ are fixed parameters. The second set $\mathbf{c}$ are referred to as context weights and can change rapidly:

$$o = \mathcal{N}(\mathbf{w}_m, \mathbf{c}, \mathbf{i})$$

Any two context vectors $\mathbf{c} = \mathbf{c}_1$ and $\mathbf{c} = \mathbf{c}_2$ identify two unique networks and thus also identify, in general, two unique policies.
Suppose we split the parameters $\mathbf{w}$ into two sets. The first set $\mathbf{w}_m$ are fixed parameters. The second set $\mathbf{c}$ are referred to as context weights and can change rapidly

$$o = N(\mathbf{w}_m, \mathbf{c}, i)$$

Any two context vectors $\mathbf{c} = \mathbf{c}_1$ and $\mathbf{c} = \mathbf{c}_2$ identify two unique networks and thus also identify, in general, two unique policies.
CONTEXT DISCERNMENT

Given a stream of data originating externally and internally of an agent, within what context is the agent? (i.e. what policy should be used?)

(AI Parlance: Given a stream of data originating externally and internally of an agent which frame am I in.)
CONTEXTUAL REINFORCEMENT LEARNING

INPUTS

Context Discerner

Rewards & Punishment

ACTION

$N(w_m, c, i)$

Manifold
Main Points

• To Describe The Architecture And Training Methodology
• To Discuss What Is Mean By Context And Contextual Discernment
• To Demonstrate This Methodology Using The Pole-Cart System
Architecture Of A Contextually Discerning Controller

\[ R(t) \xrightarrow{C_D(t)} C_{D}(t) \xrightarrow{\text{Contextually Aware Controller}} u(t) \]
What Is Meant By Context?

- **Conceptual Definition**
  - A Set Of Parameters Capable Of Representing Changes In The Dynamics Of A Plant

- **Mathematical Definition**
  - A Point On A Manifold Of Functions That Can Be Indexed By A Specific Point In An Associated Continuous Coordinate Space
The Set Of Sine Functions Specified By Coordinates
Corresponding To Amplitude, Frequency, And DC-Offset

0.5*\sin(0.7*x)+0.3
• To Discern The True Coordinates Of A Function That We Know Lies On A Specified Manifold But Has Unknown Manifold Coordinates
Contextual Discernment System

\[ D(t) = y_A(t) - y_D(t) \]

\[ x(t) \]

\[ C_D(t) \]

\[ z^{-1} \]

Function On Manifold At Location \( C_A \)

CDN (Context Discerning Network)

Function On Manifold At Location \( C_D \)

\[ \Delta C_D(t) \]

\[ y_D(t) \]
Training The Contextual Discernment System

\[ D(t) = y_A(t) - y_D(t) \]

\[ U(t) = (y_A(t) - y_D(t))^2 \]

\[ \Delta C_D(t) = u(t) \]

\[ C_D(t) = R(t) \]

\[ C_D(t) + \Delta C_D(t) = R(t+1) \]

Used To Train Critic
Training The Critic and CDN With DHP

- The Approach Is Similar To That For Training A Controller With DHP, But With The Substitutions:
  - $R(t) = C_D(t)$
  - $u(t) = \Delta C_D(t)$
  - $R(t+1) = C_D(t) + \Delta C_D(t)$

- Action Training Signal:
  - $\lambda(t+1)$

- Critic Training Signal:
  - $\lambda(t) - \frac{dU(t)}{dC_D(t)} + \lambda(t+1)(I + \frac{d\Delta C_D}{dC_D})$
Training The CDN To Discern Plant Context

\[ D(t) = y_A(t) - y_D(t) \]

\[ x(t) = z^{-1} \]

\[ u(t) \]

\[ C_D(t) \]

\[ R_A(t) \]

\[ \lambda(t) \]

\[ \Delta C_D(t) \]

\[ U(t) = (R_A(t+1) - R_D(t+1))^2 \]

\[ R_D(t+1) \]

\[ y_A(t) \]

\[ y_D(t) \]

\[ C_D(t) \]

\[ x(t) \]

\[ \lambda(t) \]

\[ \text{Used To Train Critic} \]
Context And State Variables For Training CDN Using An Analytic Pole-Cart Plant

- **C(t)**
  - Mass Of Cart Ranged From [3 5]kg
  - Length Of Pole Ranged From [0.5 3]m

- **R(t)**
  - Position Of Cart Ranged From [-2 2]m
  - Velocity Of Cart Ranged From [-2 2]m/s
  - Angle Of Pole Ranged From [-π/2 π/2]rads
  - Angular Velocity Ranged From [-2 2]rads/s

- **u(t)**
  - Control Ranged From [-5 5]Newtons
Results Of Contextual Discernment Testing

- Discerned Mass
- Discerned Length
- Actual Mass
- Actual Length

- Displacement
- Velocity
- Angular Displacement
- Angular Velocity

Squared Error

Test Iteration
Architecture Of A Contextually Discerning Controller

R(t) → Contextual Discerner → C(t) → Contextually Aware Controller → u(t)
Designing A Contextually Aware Controller

• Train A Controller Using DHP In The Standard Fashion Except…

• Add The Contextual Variables For The Plant To The State Vector Used In The Training Process

• Vary These Contextual Variables Through The Training Process
Architecture Of A Contextually Discerning Controller

\[ R_A(t) \rightarrow u(t-1) \rightarrow z^{-1} \rightarrow \text{Contextual Discerner} \rightarrow C_D(t) \rightarrow \text{Contextually Aware Controller} \rightarrow u(t) \]
Future Tasks

- Improve Contextual Discerner For The Pole-Cart System
- Train A Contextually Aware Controller For The Pole-Cart System
- Test The Ability Of The Contextually Discerning Controller In A Simulated “Real-Time” Environment
CRL Seems to Scale

Weak Convergence (avg sq err < 0.05) of CDNN, Out of 10 Trials
2,000 Training Epochs, 0.05 LR, 24 Critic/CDN Hidden Nodes
AIBO robotic dog will learn to discern and adapt to differences in walking surface types (carpet, ice, hardwood, inclines, etc.)
What I Just Told You

• The RL/ADP has focused on creating optimal policies for specific problems
• In AI parlance this is like creating a frame or a schema
• ...and, unfortunately, means we must face the Frame Problem
• ...but, fortunately, RL/ADP approaches provide a means to solve this Problem
Further Reading

All background reports/papers and upcoming Technical Report may be found at

www.nwcil.pdx.edu/pubs