Verve

Research Overview

Tyler Streeter Spring 2005

Contents

- The Problem
- Previous Work
- Current Direction
- The Verve Project
- Future Plans

The Problem

- Simulated creatures and robots cannot adapt easily to complex, changing environments
- Most current approaches use static, "hand-designed" motor control mechanisms
- An ideal agent would:
  - Learn motor control from direct experience at least as well as animals and humans
  - Learn from a simple reward signal, not from explicit feedback (i.e. critic vs. teacher)

Previous Work

- Simulated humans – standing, jumping, and walking tasks
- Learning agents represented as artificial neural networks
- Genetic algorithms (GAs) for "training"
  - A fancy hill-climbing algorithm
  - Can be used to search for good neural network parameters
- Training a neural network with a GA
  - Start with a "population" of random neural networks
  - Evaluate each one on some task
  - Throw away the bad neural networks
  - Mate the good networks to produce offspring
  - Randomly mutate the new offspring
- Videos – simulated human standing, jumping

Previous Work

- NEAT algorithm (Ken Stanley, UT)
  - Principled crossover method using "historical markings" to keep track of which genes are compatible
  - Speciation, making use of the historical markings to measure diversity
  - Incremental growth from minimal structure, ensuring a search through the smallest fitness landscape; new structure only stays when it is beneficial
- Videos – simulated biped walking

Current Direction

- Genetic algorithms worked ok for offline-training, but they don't seem biologically-realistic
- GAs require an unnatural iterative, trial-based process, but the real world contains just one long trial
- A better solution would:
  - Be more biologically-realistic
  - Have a good mathematical foundation
- Why is biological realism important?
  - Biological brains have already proven themselves as efficient learning mechanisms
  - Copying biological learning mechanisms seems to be a good route to take

Current Direction

- Reinforcement learning: “learning what to do so as to maximize a numerical reward signal”
- Strong mathematical foundation
- 3 essential components:
  – Policy: maps states to actions
  – Reinforcement signal: provides evaluative feedback
  – Value function: stores a “value” for each state
- Good agents must:
  – Try to learn the optimal value function
  – Use the value function to improve its policy
- “Reinforcement Learning” by Sutton & Barto

Current Direction

- Reinforcement learning's roots
  - Dynamic programming
    - Given a perfect model of the environment, compute the optimal policy (think IBM's Deep Blue)
    - Basically searching through all possible future states
    - Intractable for large (e.g. continuous) state spaces
  - Monte carlo methods
    - No model necessary
    - Learn directly from raw, sampled experience
    - Usually must wait until the end of a long sequence before learning anything
  - Temporal difference
    - Combination of dynamic programming and monte carlo
    - Computes the difference in value estimations between successive states
    - TD error = next reward + next value estimation – current value estimation
    - Only non-zero TD errors cause learning (i.e. surprising events cause learning; no learning once rewards are fully predicted)
    - Eventually, neutral stimuli predict rewards

Current Direction

- **Why neural networks?**
  - **Way too many states to keep track of internally**
  - **Neural networks can approximate complex state spaces with a few parameters**
  - **Biologically-realistic**

Current Direction

- If the reward comes after several actions, which action deserves the reward?
- Credit assignment problem
  - Structural
  - Temporal
- Eligibility traces
  - Each action leaves a decaying trace
  - Only eligible actions get reinforced

Current Direction

- Planning
  - Learning how the world works by building an internal model
  - Using the model to learn from “simulated experiences”
  - The better the model, the more useful the planning
  - Strongly linked to dynamic programming

Current Direction

- **Main ideas from neuroscientific research**
  - Dopamine neuron activity is somehow related to rewards
  - Most interesting hypothesis: dopamine neurons encode reward prediction errors
  - Dopamine neuron activity is <u>very</u> similar to temporal difference error signal

Current Direction

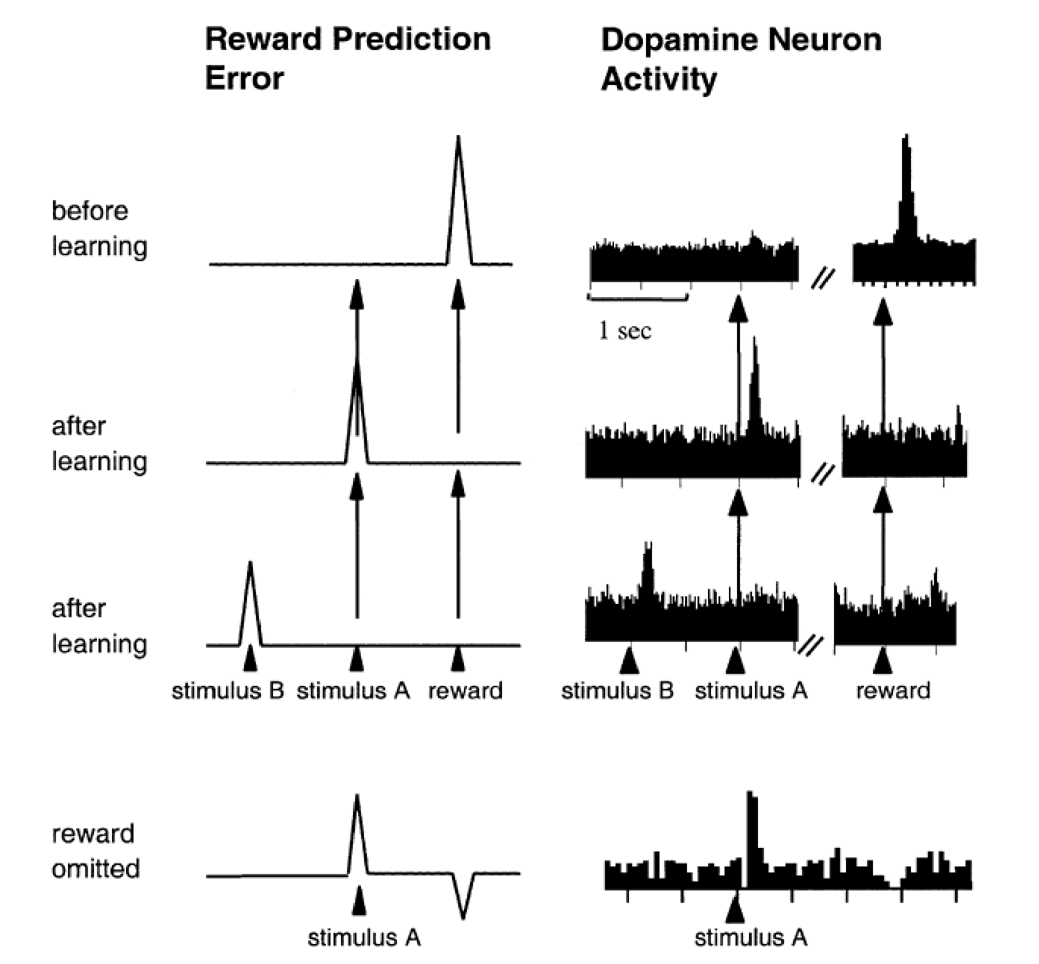

Figure taken from Suri, R. E. (2002). TD models of reward predictive responses in dopamine neurons.

Tyler Streeter Spring 2005

What is Verve?

Verve

Open Source Behavior Creation

- Reinforcement learning for general motor control tasks
  - Ground and air vehicles
  - Wheeled and legged robots/artificial creatures
  - Any controlled system with complex behaviors
- Real and simulated agents
- Biologically-inspired methods
  - Artificial neural networks for function approximation
  - Reward prediction mechanisms
- Open Source software
- Current status
  - Finishing background research in neuroscience and machine learning
  - Testing new reinforcement learning algorithms on benchmark tasks

Tyler Streeter Spring 2005

Future Plans

- Creating sample applications
  - Cart-pole test
  - Mountain-car test
  - Simulated creatures
- Planning/simulated experiences
- Train agents in simulation, then transfer them to real robots
- SETI @Home-like capabilities to distribute computations

Check the Verve website for updates:

www.vrac.iastate.edu/~streeter/verve/main.html

Tyler Streeter Spring 2005