Reinforcement Learning Blackjack

CENBD 497E
-Roger Grayson
-Ben Foote

Our Goal

- Design a Blackjack game using C++ and Allegro
- Dealer based on a flat rule
- Allow for a human player Vs. a neural network
- Also allow 2 neural networks to play

Allegro

- What is Allegro?
- A crossplatform(win32/linux/mac) graphics library
- Allows for fast implementation with minimal overhead

Flat Dealer Rule

- Dealer hit/stand decision is based on an easy assumption
- If the dealer has less than 17 for a total, it hits
- If the dealer has 17 or more, it stands, unless a player has it beat
- This flat rule serves as our benchmark

Reinforcement Learning

- Requires 3 things:
  - State (s)
  - Action (a)
  - Reinforcement signal (r)
- Learning equation:
  - $Q(s,a) = Q(s,a) + \alpha(r + \gamma Q(s', a')-Q(s,a))$
  - γ is the discount size
  - α is the learning rate
  - s' is the next state
  - a' is the next action

Interactive Vs. Automated Mode

- Can act as a real poker game with a human player
- Can also let two neural networks with different alpha and delta values play
- Automated mode gives insight to neural network performance Vs. flat rule.

Demonstration

- Any questions?

Resources

- http://www.allegro.cc
- http://www.google.com