Reinforcement Learning Algorithms used in Tic Tac Toe

December 13, 2019

1 AlphaZero

1.1 Monte-Carlo Tree Search (MCTS)

1.1.1 Upper Confidence Bound

$$U(s,a) = Q(s,a) + \sqrt{\frac{2 \ln \sum_b N(s,b)}{1+N(s,a)}}$$

$U(s,a)$ is the upper confidence bound for the current state s and action a
$Q(s,a)$ is the expected reward by taking action a in state s
$N(s,a)$ is the number of times we took action a from state s
$\sum_b N(s,b)$ is the total number of plays from state s

1.1.2 Upper Confidence Bound Alpha Zero

$$U(s,a) = Q(s,a) + c_{puct} P(s,a) \frac{\sqrt{\sum_b N(s,b)}}{1+N(s,a)}$$

$U(s,a)$ is the upper confidence bound for the current state s and action a.
$Q(s,a)$ is the expected reward by taking action a in state s.
c_{puct} is a constant that controls the amount exploration
$P(s,a)$ probability to take action a in state s as predicted by the neural network
$N(s,a)$ is the number of times we took action a from state s
$\sum_b N(s,b)$ is the total number of plays from state s

1.1.3 Alpha Zero Tree Search

procedure SEARCH(s)
  If terminal(s_t) **Then**
    Return r_t
  End If

  If not exists($P(s,.)$) **Then**
    predict $P(s,.)$ and $v(s)$ with the neural network
    $N_s(s) = 0$
    $Q(s,a) = 0$ for all a
    $N(s,a) = 0$ for all a
    If player == BLACK **Then**
      Return $-v(s_t)$
    Else
      Return $v(s_t)$
    End If
  End If

  $U(s,a) = Q(s,a) + c_{puct} P(s,a) \frac{\sqrt{N_s(s)}}{1+N(s,a)}$ for all a
  $a_t = \text{argmax}_a U(s,a)$
  Execute a_t to get next state s_{t+1}
  $v(s_{t+1}) = \text{SEARCH}(s_{t+1})$

  If player == BLACK **Then**
    $v' = -v(s_{t+1})$
  Else
    $v' = v(s_{t+1})$
  End If

  If s it root node
    $P(s) = (1-\epsilon)P(s) + \epsilon\eta_d(\alpha)$
  End If
  $Q(s,a) = \frac{N(s,a)Q(s,a)+v'}{N(s,a)+1}$
  $N(s,a) = N(s,a) + 1$
  $N_s(s) = N_s(s) + 1$
  Return v
End procedure

procedure MCTSAZ(s)
  For simulation = 1, M **Do**
    SEARCH(s_t)
  End
  Return $N(s,a)$
End procedure

1.1.4 Training Algorithm

For episode = 1, M **Do**
  For t = 1, T **Do**
    Initialize N_s, N, Q, U and P
    Initialize a fresh game
    $N(s_t, a) = \text{MCTSAZ}(s)$
    If $temp == 0$ **Then**
      $a_t = \text{argmax}_a N(s, a)$
      $P(s_t, a) = \begin{cases} 1 & \text{for } a_t \\ 0 & \text{otherwise} \end{cases}$
    Else
      $P(s_t, a) = N(s, a)^{\frac{1}{temp}}$
      $P(s_t, a) = \frac{P(s,a)}{\sum_b P(s_t,b)}$
    End If

    Append the training example $(s_t, P(s_t, a), v_t)$ to L, where v_t is arbitrary
    Pick action a_t by sampling from $P(s_t, a)$
    Play move a_t
  End

  Observe the final reward r_T of the game
  For training example $(s_t, P(s_t, a), v_t) \in L$ **Do**
    If player == WHITE **Then**
      Update $v_t \leftarrow r_T$
    Else
      Update $v_t \leftarrow -r_T$
    End If
  End
End