Algorithm 1 Deep Q-learning with experience replay

Input: $r_{\text{learn}}, n_{\text{rmem}}, \epsilon_{\text{final}}, l_{\text{expl}}, f_{\text{learn}}, f_{\text{update}}, \gamma, n_{\text{batch}}, M$

Replay Memory D を初期化
Q-Network Q をランダムな重み θ で初期化
Target network Q^- を重み $\theta^- = \theta$ で初期化
for episode $= 1, \cdots, M$ **do**
  $t = 1$
  while not *done* **do**
    ε -greedy に従って行動 a_t を選択
    $\epsilon = \max(\epsilon_{\text{final}}, \epsilon - \frac{1-\epsilon_{\text{final}}}{l_{\text{expl}}}) \to \varepsilon$ を線形減少
    行動 a_t を実行し，報酬 r_t と次の画面 x_{t+1} と *done* を観測
    前処理して次の状態 s_{t+1} を生成
    D に $(s_t, a_t, r_t, s_{t+1}, done)$ を追加，$|D| > n_{\text{nmem}}$ なら古いものを削除する．
    if $t > n_{\text{rpstart}}$ **then**
      if $(t-1)\%f_{\text{learn}} = 0$ **then**
        D からランダムに $(s_j, a_j, r_j, s_{j+1}, done)$ を n_{batch} 個の履歴をサンプル
        $$y_j = \begin{cases} r_j & (done) \\ r_j + \gamma \max_{a'} Q^-(s_{j+1}, a'; \theta^-) & (\text{otherwise}) \end{cases}$$
        θ を $(y_j - Q(s_j, a_j; \theta))^2$ の勾配方向に学習率 r_{learn} で更新（勾配計算の後の更新の際，r_j は $[-1, 1]$ にクリップされる）
      end if
      if $(t-1)\%f_{\text{update}} = 0$ **then**
        $Q^- = Q$
      end if
    end if
    $t = t + 1$
  end while
end for

Algorithm 2 Environment wrapping Atari Games

Input: $n_{\text{batch}}, n_{\text{rmem}}, l_{\text{history}}, f_{\text{tnupdate}}, \gamma, r_{\text{learn}}, m_{\text{gradient}}, m_{\text{sgradient}}, g_{\text{min}}, \epsilon_{\text{first}}, \epsilon_{\text{final}}, l_{\text{expl}}, n_{\text{rpstart}}, l_{\text{nomax}}$

Replay Memory D を初期化
Q-Network Q をランダムな重み θ で初期化
Target network Q^- を重み $\theta^- = \theta$ で初期化
for episode $= 1, \cdots, M$ **do**
  $T \sim U(1, l_{\text{nomax}})$
  for $t' = 1, \cdots, T$ **do**
    a'_t = (do nothing) の実行→初期状態の生成
  end for
  $t = 1$
  while not *done* **do**
    if $t \leq n_{\text{rpstart}}$ **then**
      a_t をランダムに決定→ Replay Memory の確保
    else
      if $(t-1)\%l_{\text{history}} = 0$ **then**
        ε -greedy に従って行動 a_t を選択
      else
        $a_t = a_{t-1}$
      end if
      $\epsilon = \max(\epsilon_{\text{final}}, \epsilon - \frac{\epsilon_{\text{first}} - \epsilon_{\text{final}}}{l_{\text{expl}}}) \rightarrow \varepsilon$ を線形減少
    end if
    行動 a_t を実行し，報酬 r_t と次の画面 x_{t+1} と *done* を観測
    前処理して次の状態 s_{t+1} を生成
    D に $(s_t, a_t, r_t, s_{t+1}, done)$ を追加，$|D| > n_{\text{nmem}}$ なら古いものを削除する．
    if $t > n_{\text{rpstart}}$ **then**
      if $(t-1)\%l_{\text{history}} = 0$ **then**
        D からランダムに $(s_j, a_j, r_j, s_{j+1}, done)$ を n_{batch} 個の履歴をサンプル
        $$y_j = \begin{cases} r_j & (done) \\ r_j + \gamma \max_{a'} Q^-(s_{j+1}, a'; \theta^-) & (\text{otherwise}) \end{cases}$$
        θ を $(y_j - Q(s_j, a_j; \theta))^2$ の勾配方向に RMSProp$(r_{\text{learn}}, m_{\text{gradient}}, m_{\text{sgradient}}, g_{\text{min}})$ を用いて更新（勾配計算の後の更新の際，r_j は $[-1, 1]$ にクリップされる）
      end if
      if $(t-1)\%(f_{\text{tnupdate}} \times l_{\text{history}}) = 0$ **then**
        $Q^- = Q$
      end if
    end if
    $t = t + 1$
  end while
end for