

FITTED-Q-LEARNING( $\mathcal{A}, s_0, \gamma, \alpha, \epsilon, m$ )

```
1   $s = s_0$  // Or draw an  $s$  randomly from  $\mathcal{S}$ 
2   $\mathcal{D} = \{ \}$ 
3  initialize neural-network representation of  $Q$ 
4  while True:
5       $\mathcal{D}_{\text{new}}$  = experience from executing  $\epsilon$ -greedy policy based on  $Q$  for  $m$  steps
6       $\mathcal{D} = \mathcal{D} \cup \mathcal{D}_{\text{new}}$  represented as  $(s, a, r, s')$  tuples
7       $\mathcal{D}_{\text{sup}} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}$  where  $\mathbf{x}^{(i)} = (s, a)$  and  $\mathbf{y}^{(i)} = r + \gamma \max_{a' \in \mathcal{A}} Q(s', a')$ 
8          for each tuple  $(s, a, r, s')^{(i)} \in \mathcal{D}$ 
9      re-initialize neural-network representation of  $Q$ 
10      $Q = \text{supervised\_NN\_regression}(\mathcal{D}_{\text{sup}})$ 
```