|  |  |  |  |
| --- | --- | --- | --- |
| Model | Principle | Pros | Cons |
| *Q-learning* | -Stores Q-values in a big matrix  -Updates Q(s,a) for visited pairs | Simple and fast to implement | -Requires **huge storage**  -**Very slow** to converge  -Requires discrete and |
| *Sarsa with global approx.* | -Stores in matrix  -Uses **eligibility trace** to propagate rewards quicker  -Uses **global approximation** to generalize to new | -Works for continuous  -**Quicker** to train than basic | -Still requires the storage of matrix |
| *Neuro-fitted Q-learning* | -**Use an MLP** to model  -Uses visited pairs to update weights | -Works for continuous  -**Requires very little storage**  -Fast to train | -Training very dependent on initialization  -Needs a tuned reward to work |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | #states  #actions | Finished 2 days ? | Best result over 2 test days | Corresponding  parameters |
| *Q-learning* | 2 400 000  15 | No | None |  |
| *Sarsa with global approx.* | 75 000  15 | Yes | 1400 $ |  |
| *Neuro-fitted Q-learning* | Continuous states  64 | Yes | **7400 $** | [32,32,16] with ReLu |

Sample variable and assume

and are piecewise uniform (using

B bins for both distributions)