HW. 5 NN and Deep Learning

Federico Bottaro

February 18, 2020

Abstract

In this document we are going to explore the field of reinforcemet learning. In particular we are dealing with a sort of game in which we can move in a 10×10 grid and the goal is to reach a fixed point starting from a random one. We are going to implement both SARSA and Q-learning and test those approach individually or combinating them.

1 Project development

The scope of our work in this document is study the combination of parameters that better perform. The important parameters that we focus on are:

- the discount factor γ that represent how much importance we are giving to future reward (discussed later). A smaller γ make our agent to give more importance to immediate rewards while γ closer to one means that the agent will give more importance to long term rewards.
- α that in this context has the meaning of a learning rate.
- The number of episodes that we perform with the same agent.
- ϵ that is a parameter that influences the next action taken by the agent. A huge value of this parameter means that the agent will take a random action.

For the evaluation of the exploration made by the agent we introduce the concept of reward appeared above. This concept consists of giving a score to the motion made by the agent. In particular we give a reward of 1 if the agent is in the goal, a reward of 0 if the agent is moving into an allowed slot and a reward of -1 if the slot is not allowed. In our first implementation a slot not allowed is just the bound of the grid. Of course if the next action is moving out of bound, we give a bad reward and we do not update the state of the agent.

2 Exploration of parameters space

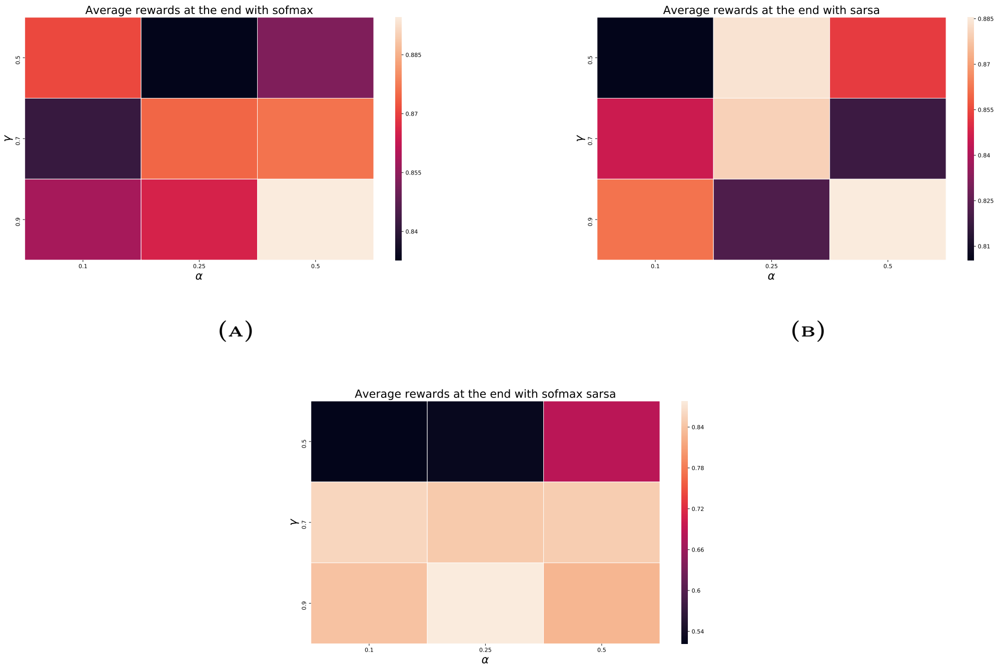
We make some assumptions before exploring the entire parameters space. For ϵ we decide to decrease the value of this parameter during the episodes. To motivate this we believe that in the early stage the agent has to explore the grid while at the end he has to make thoughtful movements. We start from a value of 0.8 until reach 0.001. For α and γ we make some preliminary research and we find out that the best order of magnitude for both the parameters is 10^{-1} so we make a grid search in this range. Using these set up we have that 3000 episods are enough to reach convergence. We try to make all the combination of parameters in three different scenario:

- We sample from a softmax distribution the probability for the next action (we call this scenario softmax).
- We use a SARSA approach to evaluate the action (we call this scenario sarsa)
- We use both the methode (called softmax sarsa).

For the evaluation of the parameters we look two different quantities:

- The mean of the reward fo the last 10 episodes
- The time (in terms of episodes) that the agent takes to reach convergence. Here for the convergence we implement a check at each episodes of the value of the rewards; if the present and the last three rewards are above 0.7 we accept the convergence and we remember the present value of the episode.

Figure 1 show the grid search using $\alpha = [0.1, 0.25, 0.5]$ and $\gamma = [0.5, 0.7, 0.9]$. Note that we present in this document results using only those value so we do not specify again the values used. As we can see we reach with all the models a good rewards mean (0.885 using sarsa alone and softmax alone and 0.84 using both the approach). A common feature for all the three model is $\gamma = 0.9$ that correspond to the best rewards. The best learning rate change depending on which model we are considering.



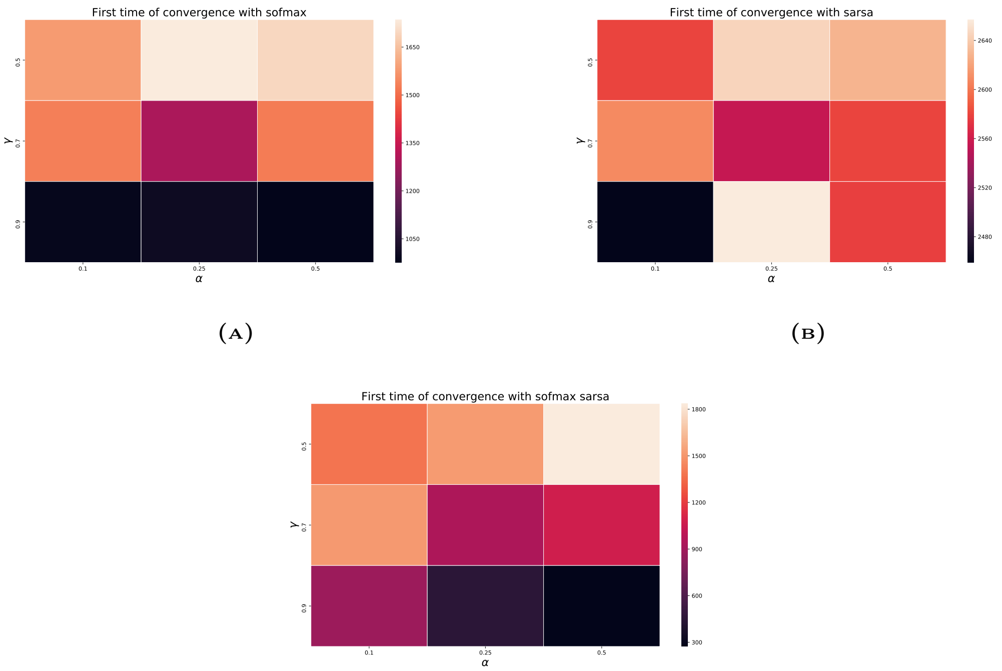
(a)

(b)

(c)

Figure 1: *Grid search for γ, α evaluating the mean value of the last ten rewards.*

Figure 2 show the behavior of the time of convergence. We can notice that the combinations of γ, α that correspond to the lower time are in the same region of the previous plot so we can conclude that these two evaluation are consistent each other.



(A)

(B)

(C)

Figure 2: *Grid search for γ, α evaluating the time of convergence.*

Considering both the grid search we decide to use as best parameters:

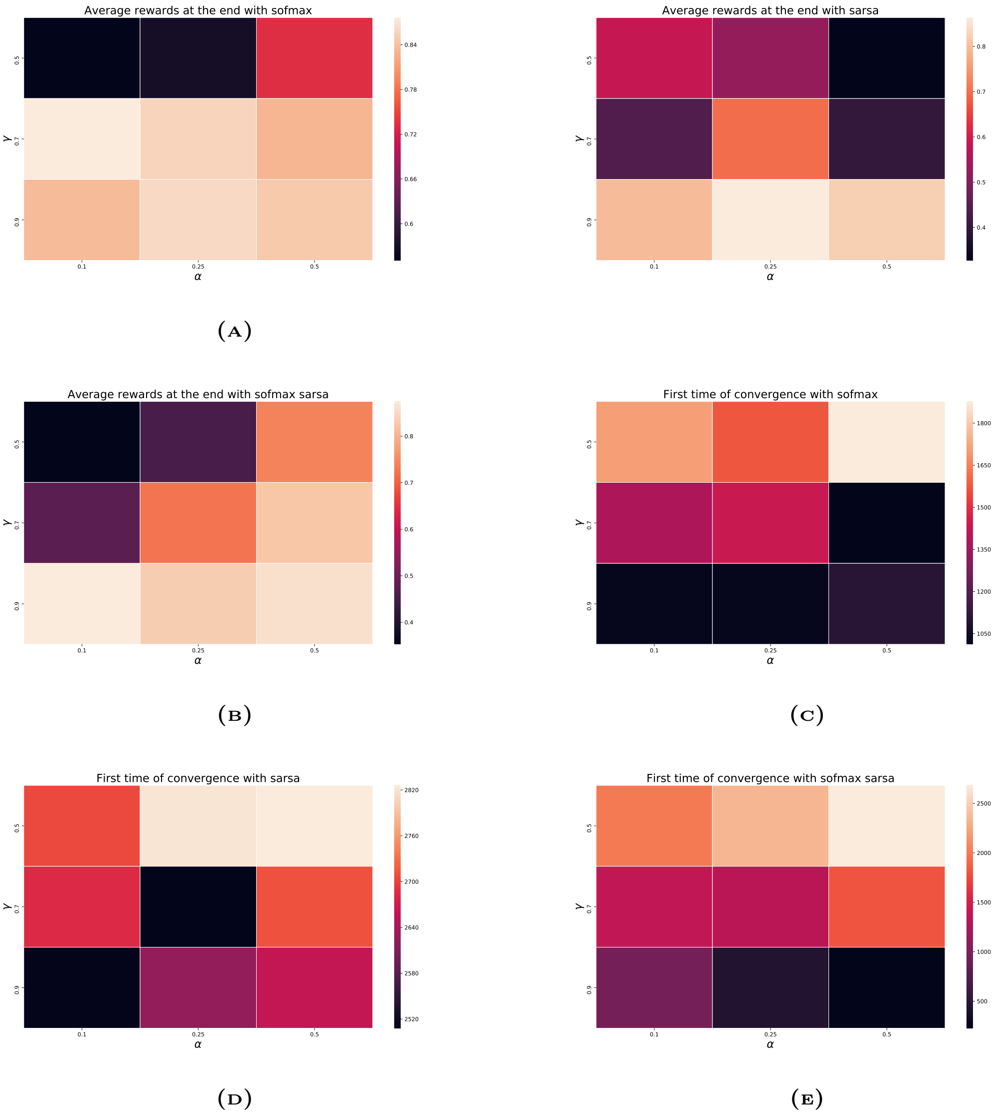
- α=0.5
- γ=0.9

and the best model is using both softmax and sarsa if we want a faster model; otherwise if we look at the mean value of the final rewards we pick up a model with only the softmax.

3 Results with more complexity

For evaluate our approach to Q-learning and SARSA we decide to train our algorithm also with a more complex grid. In particular we add to the standard 10×10 grid some wall (blocks that cannot be crossed) and some swampy blocks (that can be crossed but with a negative reward). In particular we assign a reward of -1 if an action drives the agent to a wall and we do not update the state while we assign a reward of -0.5 if an action drives the agent to a swamp but we update the state. In this case the initial points are random but we impose that the agent must start in a normal point.

The same grid search are performed and figure 3 show the results



(a)

(b)

(c)

(d)

(e)

Figure 3: *Grid search for γ, α evaluating both the mean rewards and the time of convergence.*

Also in this case we have a behavior quite close to the previous one with a general lowering of the values of final rewards and an increasing trend for the time (justifiables by the addition of complexity). We can find out the same conclusion of before: $\gamma = 0.9$ are the best and the best α is $0.25/0.5$ depending on the model. Again using both softmax and sarsa we have a faster model.

To conclude this document we print a run at the end of the training of our agent with a starting point set to $[9, 9]$. As we can see the agent try to avoid swamp and wall.

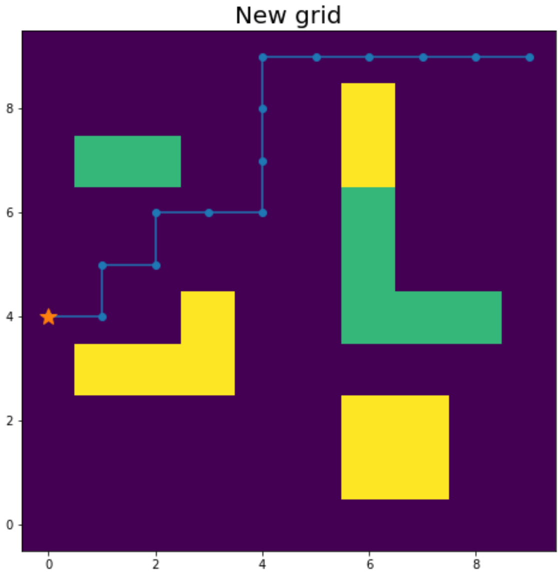


Figure 4: *Example of run of the agent once that he is trained in a complex grid. Yellow blocks are wall and green blocks swamp.*