Matching ánd Maximizing? A neurally plausible model of stochastic reinforcement learning

Jered Vroon [a] Iris van Rooij [ab] Ida Sprinkhuizen-Kuyper [ab]

[a] *Department of Artificial Intelligence, Radboud University Nijmegen*
[b] *Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour*

Abstract

An influential model of how reinforcement learning occurs in human brains is the one pioneered by Suri and Schultz [6]. The model was originally designed and tested for learning tasks in deterministic environments. This paper investigates if and how the model can be extended to also apply to learning in stochastic environments. It is known that if rewards are probabilistically coupled to actions that humans tend to display a suboptimal type of behavior, called *matching*, where the probability of selecting a given response equals the probability of a reward for that response. Animal experiments suggest that humans are unique in this respect. That is, non-human animals display an optimal type of behavior, called *maximizing*, where responses with the maximum probability of a reward are consistently selected. We first show that the model in its original form becomes inert when confronted with a stochastic environment. We then consider two natural adjustments to the model and observe that one of them leads to matching behavior and the other leads to maximizing behavior. The results yield a deeper insight in the workings of the model and may provide a basis for a better understanding of the learning mechanisms implemented by human and animal brains.

When we enter the world as infants there are many things we cannot do yet, but as the years pass our behavioral repertoire increases vastly. For example, we learn to sit, to walk, to talk, to read, etc. Understanding the human ability for *learning* seems crucial for understanding how we come to display all kinds of adaptive and intelligent behaviors. Such an understanding can also be put to practical use in the context of artificial intelligence, as it may afford building machines that can learn all that humans can learn. In this paper we focus on a specific, yet common, form of learning called *reinforcement learning*. Reinforcement learning is a type of learning where the learner is given minimal information about his or her performance on the task that has to be learned. Feedback is given on preformed actions, but no feedback is given about what feedback other actions would have yielded. In the task used in this paper the only feedback given is whether the given action was correct or not.

In 1998, Schultz discovered a systematic relationship between the activity of dopamine neurons and reinforcement learning [5]. Soon after, Suri and Schultz used these insights to propose a biologically inspired model of reinforcement learning [6]. These authors were among the first to propose a model of this type, but see also for example [2]. Even though many advances have been made in this field since, the model of Suri and Schultz incorporated many of the general principles that are still used to date. Also, the model is less complicated than many of its successors. These two properties make the model well suited for testing the essence of this class of models. Suri and Schultz [6] trained their model on a deterministic task with delayed rewards. Not only was the model capable of learning to perform the task, but Suri and Schultz observed that the model followed a learning curve similar to that of monkeys and activations of key components in the model qualitatively matched the pattern of neural activity in the monkey's basal ganglia. These results are impressive and show that the neurophysiological mechanisms underlying reinforcement learning in the brain can be captured in computational models. One important limitation, however, is that the results were attained specifically using a deterministic task (i.e., a task in which an action is guaranteed to yield the same feedback every time it is performed). The real world is inherently uncertain. Almost every action we can perform will sometimes be successful and sometimes not. A stronger test of the model, and its ecological validity, could thus be achieved if we test its performance on a stochastic task.

In a stochastic task each action has some probability $0 < p < 1$ of being successful. In this situation it is nearly impossible to select actions that are always successful. In the literature, two qualitatively different strategies have been reported, where one is known to be specifically associated with human performance and the other with animal performance. Humans tend to use a suboptimal strategy, called *matching*, which consists of selecting a given action with a probability that equals the probability of a reward for that action. In contrast, non-human animals tend to use an optimal strategy, called *maximizing*, which consists of always selecting the action with the maximum probability of a reward. Given this known characteristic difference between humans and non-human animals in how they perform on a stochastic task, it is of interest to investigate if the model of Suri and Schultz can model either one of these strategies or possibly both. By investigating what settings cause the model to exhibit these strategies we may gain a deeper understanding of the reinforcement learning mechanisms operational in human and animal brains.

The remainder of this paper is organized as follows. We start, in Section 1, with a description of the model. Section 2 presents details of the task and the strategies used by humans and animals. Next we report on the results of our simulations. We show how, and explain why, the model in its original form becomes inert when trained on the stochastic task (Section 3). We consider two classes of adaptations to the model, both sufficing to overcome the initial inertia. We observe that the first class of adaptations causes the model to display the matching strategy (Section 4.1), and that the second class of adaptations causes the model to display the maximizing strategy (Section 4.2). We conclude by discussing the significance and implications of our findings in Section 5.

1 Model

In this section we will discuss the relevant parts of the model by Suri and Schultz [6]. We first give a general description of the model and its parts, followed by a more specific step by step description of each of those parts. Figure 1 gives an overview of the model.

The model consists of two neural network-like components designated as the Critic and the Actor. The Actor decides which action is performed (the number of actions is limited). If the correct action is chosen at the correct time the model will be rewarded after some delay. Meanwhile, the Critic makes a prediction about the expected reward, which is compared to the actual reward, rendering the so-called Effective Reinforcement Signal. The Effective Reinforcement Signal is positive if a reward is given but not predicted, negative if no reward is given but predicted, and zero if a reward is given as predicted. This signal then is used to change the behaviour of both Actor and Critic. The chosen action is tuned so that it is less likely to be chosen next time if the Effective Reinforcement Signal is negative and so that it is more likely chosen if it is positive. If the Effective Reinforcement Signal is zero, no adaptations are made. The Critic is tuned similarly.

Input
In the model, time plays an important role. The input is an $n \times t$ matrix of 1s and 0s, where n is the number of stimuli and t is the number of time steps in the trial. A value 1 represents the presence of a stimulus, and 0 its absence. The reward is represented as a stimulus as well. For example, a trial could consist of some delay after which a certain stimulus or combination of stimuli is presented for several time steps. If the correct actions were performed at the correct moments during the trial, the stimulus representing reward is presented for a number of time steps. Generally there is a delay between the actions being performed and the reward.

The following descriptions of the other parts all describe their activity in a single time step. These activities are repeated for every time step in the trial.

Actor
In the Actor, nodes representing all possible actions, *output nodes*, are fully connected to nodes representing the current input, *input nodes*. The connections are weighted. The current values of the input nodes are weighted and summed over the connections and thus lead to a value, or *activation*, for the output nodes. Some noise is added to this activation as well. The action represented by the output node with the highest positive activation is the action chosen and executed by the model. If none of the output nodes have an activation above zero, no action is performed (unintentionally doing nothing). If two output nodes are tied for the highest activation, the model is in equilibrium and no action will be performed either. There can also

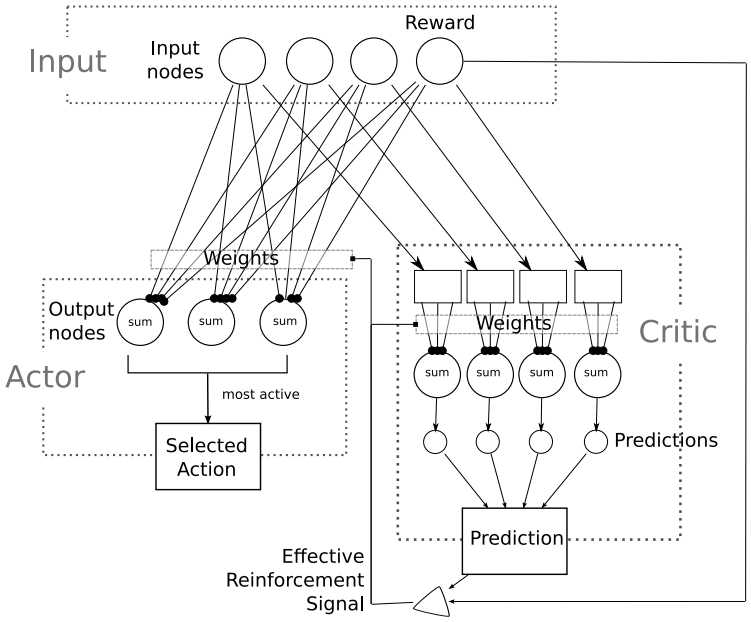

Figure 1: Overview of the model. Input is given to the model in the form of activation values on the input nodes. Activation values can be either 1 or 0, representing the presence or absence of a stimulus respectively. In the Actor, these activations are weighted and summed to determine the activation of the output nodes. Each one of these nodes is associated with one of the possible actions, and the action associated with the output node with the highest activation is the selected action. The selected action is subsequently compared with the correct action and if they match reward will be given on that trial. Meanwhile, the Critic represents the activations of the input nodes in a temporal fashion. These representations are weighted and summed to yield local predictions of the reward, which are then combined to a global prediction of the reward. The Effective Reinforcement Signal is then computed by subtracting this prediction from the actual received reward. The last step is that the weights of the Actor and Critic are updated with the Effective Reinforcement Signal.

be an action that represents doing nothing (intentionally doing nothing).

Output

The output consists of the action chosen by the Actor. When given, the output is compared to the correct output, which can change from trial to trial. If the given output does not match the correct output, the stimulus representing the reward will not be presented during this trial. In other words, the model will only be rewarded if its output was the correct output at every time step throughout the trial.

Critic

The Critic makes a prediction about which stimuli, including rewards, are expected at current time step. To do so, temporal representations of the present and recent stimuli are used. The values in these representations are weighted and summed to form a prediction of reward. This prediction is compared to the actual reward, to calculate the aforementioned Effective Reinforcement Signal. If reward is given but not predicted, the signal is positive. If reward is predicted but not given, the signal is negative. If reward is given when predicted or not given when not predicted the signal is zero.

Learning

The Effective Reinforcement Signal is then used to change the behaviour of the Actor and the Critic. If the signal was positive the chosen action is made more to be likely chosen next time by increasing the associated weights. If the signal was negative the chosen action is made less likely to be chosen next time by decreasing the associated weights. Any decrease is 6 times stronger than the increase on a positive Effective Reinforcement Signal to prevent perseverations. At the same time also the weights of the Critic are manipulated to improve the quality of the prediction. If the Effective Reinforcement Signal is zero, the action and prediction were correct and no changes are made.

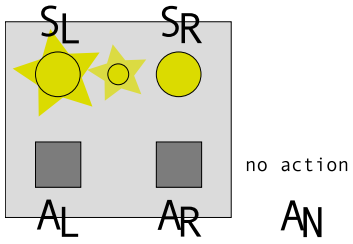

Figure 2: Graphical representation of the task. The task consists of two stimuli, S_L and S_R, which can be thought of as two lamps that are either turned on or off, and three possible actions, A_L, A_R, A_N which can be thought of as pushing the left button, the right button or doing nothing respectively. We distinguish between the model not deciding to do something (in which case no action is selected) or decided to do nothing (in which case the 'action' A_N is selected). The option of selecting A_N, was included to achieve a better fit with the original model. A third stimulus that is 'turned on' whenever one of the other two stimuli is 'turned on' was included for the same reason.

2 Task

To stay as true as possible to the original study by Suri and Schultz, our stochastic task is modeled after their deterministic task, with the only difference being that the mapping between stimuli and correct actions is stochastic. The task consists of several different trials. Every trial in turn consists of 80 successive moments in time to which we refer as time steps, during which one of two target stimuli is presented (from time step 40 to 45). We refer to these two stimuli as S_L and S_R and they can be thought of as two lamps; one on the left, one on the right. As in the original task, a third stimulus is presented as well if one of these target stimuli is presented. At each and every one of these time steps the model can perform one of three 'actions'. Two of these, A_L and A_R, are easily recognized as actions and can be viewed as pressing one of two buttons; one on the left, one on the right. The other possible action A_N, doing nothing intentionally, is adopted from the original task. Another possibility is to do nothing unintentionally by not selecting any action, which occurs when none of the actions becomes active enough to be chosen and thus can be viewed as the default 'action'.

Until a target stimulus is presented, the model should perform no action, either intentionally or unintentionally. When the target stimulus is presented the correct action should be chosen, which is either A_L or A_R since doing nothing in response to a stimulus is never correct. If the model chose the correct actions at the right time steps a reward will be given to it at time steps 51 and 52 (which means there is a delay between action and reward).

The stochastic aspect of the task is reflected by the probabilistic coupling between the stimulus and the correct action. At the beginning of the trial the stimulus that will be presented during that trial (either S_L or S_R) is chosen at random, as well as what action will be the correct action. For each of the stimuli in p ($0.5 \leq p \leq 1$) of the trials one of the actions A_L and A_R is correct, while in $1 - p$ of the trials the other action is correct. We will refer to the action that is correct with probability p as the *consistent* action and to the action that is correct with probability $1 - p$ as the *inconsistent* action.

2.1 Maximizing

The maximizing strategy involves nothing but consequently picking the consistent action. As the name suggests, this strategy maximizes average reward, simply because picking the option with the highest probability of being rewarded has the highest probability of being rewarded. This strategy is the strategy commonly and consistently used by non-human animals in tasks like this [3]. Some preliminary observations from our own lab suggest children maximize as well.

2.2 Matching

The matching strategy entails picking the consistent action with probability p and the inconsistent action with probability $1 - p$. This strategy is commonly used by adult humans. By using the terms p and $1 - p$ we do not mean to suggest that picking the consistent or inconsistent action happens at random. On the contrary, quite elaborate theories of for example hypothesis formulation have been suggested to explain matching behaviour [7]. This theory is supported by the observations that adults come up with elaborate schemes of consistent and inconsistent actions and motivations why those schemes are good [1, 8]. Notably,

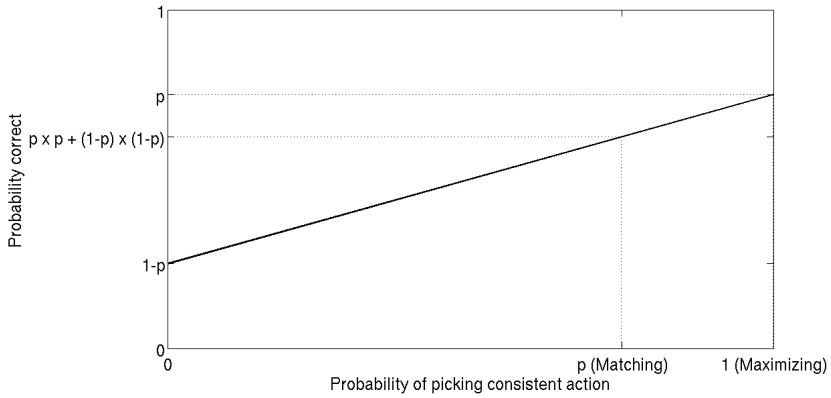

Figure 3: Illustration of the differences in performance (measured by probability correct) between matching and maximizing. Maximizing is defined as always (with probability 1) selecting the consistent action. This strategy yields the optimal average performance, i.e., being correct in p of the cases. Matching is defined as selecting the consistent action in p of the cases. This yields a suboptimal average performance, i.e., being correct in $p \times p + (1-p) \times (1-p)$ of the cases. There is a linear relationship between the probability of picking the consistent action and the average reward. A logical consequence of this is that the strategy (e.g., matching or maximizing) can be directly inferred from the proportion of correct answers if averaged over a sufficiently large number of trials.

since the task is truly stochastic no such scheme actually applies to the task. This is reflected in the average reward that this strategy yields. On average this strategy will be rewarded in $p \times p + (1-p) \times (1-p)$ of the cases. Given that $p > 1 - p$ it is easy to see that this means it will be rewarded in less than p of the cases.

2.3 Matching versus maximizing

Matching on average yields a lower probability of being correct than maximizing, which means that on average non-human animals outperform adult humans. Important for the use of this paper is that this relationship between used strategy and probability correct is linear (see Figure 3). This implies that if average performance of the model is similar to that of either of the strategies, we can conclude the model uses that strategy.

3 Performance of the original model

We simulated the performance of the model of Suri and Schultz [6] on the stochastic task. We observed that, after several trials, the model became inert. The weights between input nodes and output nodes decreased until none of the output nodes had an activation above zero, as a consequence of which none of the actions was selected anymore (not even A_N). This in turn led to a drastic decrease in the model's performance on the task (see column 4 of Table 1). We next explain, in conceptual terms, how this inadequate performance of the model can be seen as an artifact of it being optimized for a deterministic task.

In the original model, the decrease in weights after having received no reward was much higher (6 times) than the increase in weights after having received a reward. This parameter setting was used by Suri and Schultz to prevent perseverations on erroneous actions. This setting optimizes performance in a deterministic task, because in the context of a deterministic task the absence of a reward is a reliable cue for not performing that action when presented with the stimulus. In a stochastic task, on the other hand, an action may well not yield a reward merely due to bad luck. Performing the same action in response to the same stimulus may on a different trial lead to a reward. Moreover, the possibility of not being rewarded once in a while holds even for the optimal strategy for the stochastic task; i.e., maximizing will yield a reward with probability p, not 1. The large decrease of weights on trials where rewards are absent, and the smaller increase of weights on trials where the rewards are present, causes Actor weights for any given action to overall decrease faster than they increase. After several trials, this decrease of the weights causes none of the actions is activated above zero anymore. The model does not respond anymore. In other words, the model shows general inertia.

p	Maximizing	Matching	Original Model	Overcoming Inertia		Preventing Inertia
				Forced Action	Threshold	
1	1.00	1.00	1.00	1.00	1.00	1.00
0.9	0.90	0.82	0.03	0.85	0.77	0.89
0.75	0.75	0.63	0.01	0.66	0.58	0.74
0.6	0.60	0.52	0.00	0.53	0.44	0.59

Table 1: Average performance of maximizing and matching strategies, the original model, and our adaptations to the model for several values of p. Averaged performance of maximizing and matching was calculated. Average performance of our adaptations on the model was averaged over trial 100-500 from 5 runs. This is because we observed the model to generally have settled after about 100 trials.

In sum, the model in its original form can simulate neither human nor animal performance on the stochastic task, as it fails to act at all. We next consider two different ways for solving the observed problem of inertia. The first way is to somehow overcome the detrimental effects of the large decrease in Actor weights (Section 4.1), and the second is to somehow prevent the weights from dropping as fast as they do in the original model (Section 4.2).

4 Two Classes of Adaptations of the model

4.1 Overcoming the inertia

There are numerous ways of overcoming the inertia of which we will discuss two representative ones. The first one is forced action; forcing the model to act even if the activations for the different actions are negative. The second is preventing the weights from dropping below a certain (positive) threshold by increasing all weights when one of them drops below the threshold.

Even though the implementational details of these solutions differ, the general idea behind them is the same. Each one is a way to undo the negative effects of constantly decreasing weights on an action being chosen.

Forced action
Forced action entails forcing the model to pick one of the actions A_L or A_R each time an action should be picked. To do so, the unintentional and intentional doing nothing are disabled. Since the unintentional inactivity is caused by either negative weights or an equilibrium, this was implemented by having the model pick the most active action even if activation is negative and by choosing at random in case of an equilibrium.

Putting a threshold on the weights
Putting a positive threshold on the weights is an elegant and biologically plausible solution. The weights do not drop below the threshold and the model does not become inactive from negative weights.

Yet, there is a catch. Merely introducing a threshold does not work. The weights then decrease to the threshold, bringing the model in a perfect equilibrium. This equilibrium is so perfect it causes inertia, which is not exactly a solution to the problem. To prevent this equilibrium, instead of cutting off all weights that drop below the threshold, we increase all weights if otherwise one of them would drop below the threshold. This guarantees the weights will not drop below the threshold and evades the equilibrium.

4.1.1 Results

We ran simulations for each of the abovementioned adaptations. The results are the same for both suggested adaptations: the model does not show inertia anymore. Instead the model shows matching behaviour (see column 5 and 6 in Table 1). The threshold yields a lower probability correct than matching and forced action for sometimes A_N is selected, which yields no reward.

The alternations between consistent and inconsistent actions that define matching behaviour arise indirectly because only the weights connected to the node representing the chosen action learn. As a consequence of this only those weights are decreased. Eventually this will cause those weights to become so low

that a different action is chosen. This same process is repeated over and over again, leading to an alternation of actions. Furthermore, as these alternations are caused by decreases of the weights, which are given with probability p for the consistent action and probability $1 - p$ for the inconsistent action, they follow these probabilities. This causes matching, a mathematical proof of which is given in the appendix.

4.2 Preventing the inertia

In this section we will discuss how the inertia can be prevented. The inertia was caused by the decrease in weights on the model not being rewarded unexpectedly being stronger than the increase in weights on the model being rewarded unexpectedly. To prevent the inertia, one simple adaptation seems to be sufficient: make the decrease in weights caused by not being rewarded less severe.

However, the decrease in weights was stronger than the increase in weights in the original model with a reason. It was introduced to prevent the model from perseverating. Without this measure the model could end up picking the wrong action if, by some coincidence, the inconsistent action is rewarded quite often in a row. Even though this effect wears of after some time, it still is not the best behaviour possible.

In an attempt to prevent the inertia without losing the advantages of having a stronger decrease than increase in weights, we did not make the strength of decrease and increase equal. Instead we opted for making the decrease only a little stronger than the increase using a ratio of 1.3, rather than the original ratio of 6. This ratio of 1.3 was chosen for it was the lowest fraction that successfully prevented most perseverations.

4.2.1 Results

We ran simulations with the adaptation in place and observed that the model does not become inert anymore. In contrast to the previously discussed adaptations that overcome the inertia, with this adaptation the alterations between the different actions stop as well. Since, with the change, correct actions are rewarded almost as much as incorrect actions are punished, instead of a race to be silent the actions now start a race to be the most active action. And since the consistent action will be rewarded most often, in most cases the model will end up picking the consistent action all the time. This behaviour is maximizing (see column 7 in Table 1).

In other words, with these changes the model shows behaviour reminiscent of that of non-human animals.

5 Conclusions

We have investigated if and how an existing biologically motivated model of reinforcement learning can be extended to apply to a stochastic task. The model, in its original form, was found to show general inertia when confronted with a stochastic task. This inertia was the result of the weights dropping so low that none of the actions was activated anymore. We considered two classes of adaptations. One of them involved overcoming the detrimental effects of the weights decreasing strongly. The other involved preventing the weights from decreasing as strong as they did. Both of these adaptations succeeded in undoing the inertia. With the first adaptation, the model displayed matching behaviour, which is the strategy used by humans. With the second adaptation, the model displayed maximizing behaviour, which is the strategy used by other animals.

Our adaptations are quite local and involve a minimal of change to the original model of Suri and Schultz [6]. Yet, they suffice for explaining two qualitatively different strategies (the one used by humans and the one by animals). Other models explain only one of these strategies, viz., matching, and require more assumptions [2]. Furthermore, our adaptation to overcome the initial inertia (implemented either as forced action or a threshold) provides a more parsimonious explanation of matching behavior than other models that assume such behavior arises from explicit hypothesis formation [7]. This is not to say that humans do not engage in hypothesis formation in a stochastic task, but this process may be an afterthought rather than the cause of the matching behaviour.

Furthermore, we have shown that matching and maximizing are quite similar to one another; by changing from one class of adaptations to another the strategy displayed by the model changes from matching to maximizing or vice versa. Though somewhat similar findings have been done before [4], our findings are new in that they are done on a neurally inspired model.

In closing, we note that our adaptations of the model have led to an acting model of reinforcement learning that applies both to deterministic and stochastic tasks, whereas the original model applied only to deterministic tasks. We have shown how the model can be adapted to display either matching or maximizing behaviour. Neuroscientific studies may determine the possible biological implementations of the adaptations that we have proposed.

6 Acknowledgement

The authors would like to thank Els van Dijk for her useful comments on earlier versions of this paper.

References

[1] W. K. Estes. A descriptive approach to the dynamics of choice behavior. *Behavioral Science*, 6:177–184, 1961.

[2] W. T. Fu and J. R. Anderson. From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General*, 135(2):184–206, 2006.

[3] J. M. Hinson and J. E. R. Staddon. Matching, maximizing and hillclimbing. *Journal of the Experimental Analysis of Behavior*, 40:321–331, 1983.

[4] Y. Sakai and T. Fukai. When does reward maximization lead to matching law? *PLoS ONE*, 3(11):e3795+, November 2008.

[5] W. Schultz. Predictive reward signal of dopamine neurons. *Journal of Neurophysiology*, 80:1–27, 1998.

[6] R. E. Suri and W. Schultz. A neural network model with dopamine-like reinforcement signal that learns a spatial delayed response task. *Neuroscience*, 91(3):871–890, 1999.

[7] G. Wolford, M. B. Miller, and M. Gazzaniga. The left hemisphere's role in hypothesis formation. *Journal of Neuroscience*, 20(6):64RC–, 2000.

[8] J. I. Yellott Jr. Probability learning with noncontingent success. *Journal of Mathematical Psychology*, 6(3):541–575, October 1969.

Appendix: proof of how switching when punished causes matching

In the simplified case the model switches action every time its current action is not rewarded (A_N is left out of the equation) we can view the models behaviour as a Markov Chain.

In this Markov Chain there would be two states, namely choosing the consistent action (C) and choosing the inconsistent action (I). If choosing the consistent action, the model is not rewarded with probability $1-p$ and has an equal probability of changing to choosing the inconsistent action. Probability of reward is p and thus we have an equal change of going on choosing the consistent action. For choosing the inconsistent action these probabilities are inversed.

It is now easy to see that the probability of the model picking the consistent action ($P(C)$) is equal to the weighted sum of the probabilities of it picking C while in C and it picking C while in I.

$$P(C) = pP(C) + pP(I)$$

This can be rewritten as $P(C) = p(P(C) + P(I))$ and since $P(C) + P(I) = 1$ (making the reasonable assumption the model always picks at least one of the actions), that in turn can be rewritten as:

$$P(C) = p$$

This means that, given the assumptions of this simplified case, the probability of choosing the consistent action is equal to p. This is matching.