PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Models that learn how humans learn: The case of decision-making and its disorders

Amir Dezfouli[1,2], **Kristi Griffiths**[3], **Fabio Ramos**[4], **Peter Dayan**[5,6‡], **Bernard W. Balleine**[1‡*]

1 School of Psychology, UNSW, Sydney, Australia, **2** Data61, CSIRO, Australia, **3** Westmead Institute for Medical Research, University of Sydney, Sydney, Australia, **4** University of Sydney, Sydney, Australia, **5** Gatsby Computational Neuroscience Unit, UCL, London, United Kingdom, **6** Max Planck Institute for Biological Cybernetics, Tübingen, Germany

‡ These authors are joint senior authors on this work.
* bernard.balleine@unsw.edu.au

OPEN ACCESS

Citation: Dezfouli A, Griffiths K, Ramos F, Dayan P, Balleine BW (2019) Models that learn how humans learn: The case of decision-making and its disorders. PLoS Comput Biol 15(6): e1006903. https://doi.org/10.1371/journal.pcbi.1006903

Editor: Jakob H Macke, Stiftung caesar, GERMANY

Received: July 6, 2018

Accepted: February 25, 2019

Published: June 11, 2019

Data Availability Statement: All data are available within the manuscript and Supporting Information files.

Funding: This research was supported by a grant from the NHMRC, GNT1089270 to BB. BB was supported by a Senior Principal Research Fellowship from the National Health and Medical Research Council of Australia, GNT1079561; https://nhmrc.gov.au. PD was partly funded by the Gatsby Charitable Foundation; http://www.gatsby.org.uk. The funders played no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

Popular computational models of decision-making make specific assumptions about learning processes that may cause them to underfit observed behaviours. Here we suggest an alternative method using recurrent neural networks (RNNs) to generate a flexible family of models that have sufficient capacity to represent the complex learning and decision- making strategies used by humans. In this approach, an RNN is trained to predict the next action that a subject will take in a decision-making task and, in this way, learns to imitate the processes underlying subjects' choices and their learning abilities. We demonstrate the benefits of this approach using a new dataset drawn from patients with either unipolar (n = 34) or bipolar (n = 33) depression and matched healthy controls (n = 34) making decisions on a two-armed bandit task. The results indicate that this new approach is better than baseline reinforcement-learning methods in terms of overall performance and its capacity to predict subjects' choices. We show that the model can be interpreted using off-policy simulations and thereby provides a novel clustering of subjects' learning processes—something that often eludes traditional approaches to modelling and behavioural analysis.

Author summary

Computational models of decision-making provide a quantitative characterisation of the learning and choice processes behind human actions. Designing a computational model is often based on manual engineering with an iterative process to examine the consistency between different aspects of the model and the empirical data. In practice, however, inconsistencies between the model and observed behaviours can remain hidden behind examined summary statistics. To address this limitation, we developed a recurrent neural network (RNNs) as a flexible type of model that can automatically characterize human decision-making processes without requiring tweaking and engineering. To show the benefits of this new approach, we collected data on a decision-making task conducted on subjects with either bipolar or unipolar depression, as well as healthy controls. The results

Competing interests: Part of this work was conducted while PD was visiting Uber Technologies. The latter played no role in its design, execution or communication.

showed that, indeed, important aspects of decision-making remained uncaptured by typical computational models and even their enhanced variants, but were captured by RNNs automatically. Further, we were able to show that the nature of such processes can be unveiled by simulating the model under various conditions. This new approach can be used, therefore, as a standalone model of decision-making or as a baseline model to evaluate how well other candidate models fit observed data.

Introduction

A computational model of decision-making is a mathematical function that inputs past experiences—such as chosen actions and the value of rewards—and outputs predictions about future actions [e.g. 1, 2, 3]. Typically, experimenters develop such models by specifying a set of structural assumptions along with free parameters that allow the model to produce a range of behaviors. The models are then fitted to the observed behaviors in order to obtain the parameter settings that make the model's predictions as close as possible to the empirical data. Nevertheless, if the actual learning and choice processes used by real human subjects differ from those assumptions, e.g., if a single learning-rate parameter is assumed to update the effects of reward and punishment on action values when they are in fact modulated by different learning-rates, then the model will misfit the data [e.g., 4]. To overcome this problem, computational modelling often involves an iterative process that includes additional analyses to assess assumptions about model behavior, subsequent emendation of the structural features of the model to reduce residual fitting error, then new analyses, and so forth. The final model is that which is simplest and misfits the least. This iterative process has been common practise for model development in domains such as cognitive science, computational psychiatry, and model-based analyses of neural data [e.g., 5, 6, 7, 8, 9, 10, 11]. This approach is, however, limited from two standpoints: (i) It is typically unclear when to stop iterating over models. This is because in each iteration the unexplained variance in the data can be either attributed to the natural randomness of humans actions, which implies that no further model improvement is required, or to the lack of a mechanism in the model to absorb the remaining variance, which implies that further iterations are required. (ii) Even if it is believed that further iterations are required, improving the model will be mostly based on manual engineering in the hope of finding a new mechanism that, when added to the model, provides a better explanation for the data.

Here to address these limitations we consider an alternative approach based on recurrent neural networks (RNNs); a flexible class of models that make minimal assumptions about the underlying learning processes used by the subject and that are known to have sufficient capacity to represent any form of computational process [12], including those believed to be behind the behaviour of humans and other animals in a wide range of decision-making, cognitive and motor tasks [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. Since these models are flexible, they can automatically characterize the major behavioral trends exhibited by real subjects without requiring tweaking and engineering. This is achieved by training a network to learn how humans learn [25, 26, 27, 28], which involves adjusting the weights in a network so that it can predict the choices that subjects make both during learning and at asymptote. At this point the weights are frozen and the model is simulated on the actual learning task to assess its predictive capacity and to gain insights into the subjects' behavior. This approach is not prone to the problems mentioned earlier because RNNs can in principle be trained to represent any form of behavioural process without requiring manual engineering; however, a potential problem is

that the models are so flexible that they may overfit the data and not generalize in a relevant manner; an issue that we address by using regularisation methods and cross-validation.

To illustrate and evaluate this approach, we focus on a relatively simple decision-making task, involving a two-arm bandit, in which subjects chose between two actions (button presses) that were rewarded probabilistically. To examine the predictive capacity of RNNs under typical and atypical conditions, data from three groups were collected: healthy subjects, and patients with either unipolar or bipolar depression. We found that RNNs were able to learn the subjects' decision-making strategies more accurately than both baseline reinforcement-learning and logistic regression models. Furthermore, we show that off-policy simulations of the RNN model allowed us to visualize, and thus uncover, the properties of the learning process behind subjects' actions and that these were inconsistent with the assumptions made by reinforcement-learning treatments. Furthermore, we illustrate how the RNN method can be applied to predict diagnostic categories for different patient populations.

Results

Model and task settings

RNN. The architecture we used is depicted in Fig 1; it is a particular form of recurrent neural network. The model is composed of an LSTM layer [Long short-term memory; 29], which is a recurrent neural network, and an output softmax layer with two nodes (since there are two actions in the task). The inputs to the model on each trial are the previous action and the reward received after taking the action, and the outputs of the model are the probabilities of selecting each action on the next trial. We refer to the framework proposed here as RNN.

The LSTM layer is composed of a set of interconnected LSTM cells, in which each cell can be thought of as a memory unit which maintains and updates a scalar value over time (shown by h_t^i in Fig 1 for the i*th* LSTM cell at time *t*). On each trial, the value of each cell is updated based on the inputs and on the last value of the other LSTM cells in the network (including the cell itself), and in this way the LSTM layer can track relevant information regarding the history of past rewards and actions. Each LSTM cell outputs its current value (h_t^i) to the softmax layer through an additional set of connections that determine the influence of the output of each cell

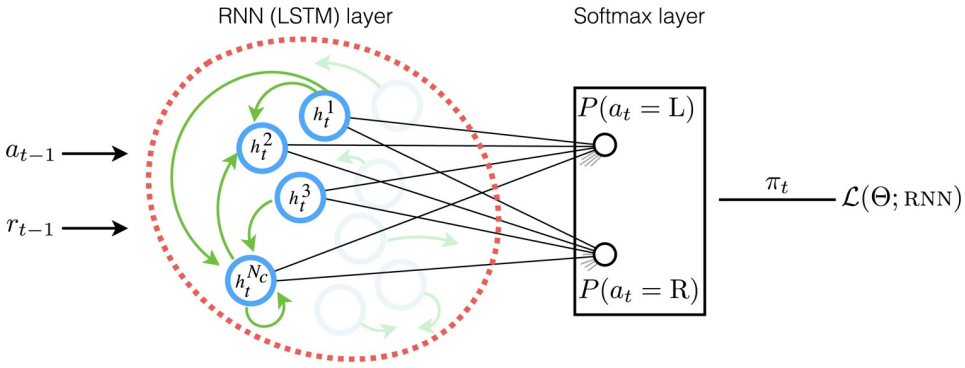

Fig 1. Structure of the RNN model. The model has an LSTM layer (shown by red dashed line) which receives the previous action and reward as inputs, and is connected to a softmax layer (shown by a black rectangle) which outputs the probability of selecting each action on the next trial (policy). The LSTM layer is composed of a set of LSTM cells (N_c cells shown by blue circles), that are connected to each other (shown by green arrows). The outpt of the cells (denoted by h_t^i for cell *i* at time *t*) are connected to a softmax layer using a set of connections shown by black lines. The free parameters of the model (in both LSTM and softmax layers) are denoted by Θ, and $\mathcal{L}(\Theta, \mathrm{RNN})$ is a metric which represents how well the model fits subjects' data and is used to adjust the parameters of the model using the maximum-likelihood estimate as the network learns how humans learn.

https://doi.org/10.1371/journal.pcbi.1006903.g001

on the predictions for the next action (shown by lines connecting them in Fig 1). As a whole, such an architecture is able to *learn* in a decision-making task by tracking the history of past experiences using the LSTM layer, and then turning this information into subsequent actions through the outputs of the softmax layer.

The way in which a network learns in the task and maps past experiences to future actions is modulated by weights in the network. Here, our aim was to tune the weights so that the network could predict the next action taken by the subjects—given that the inputs to the network were the same as those that the subjects received on the task. This is learning how humans learn, in which the weights are trained to optimise a metric (denoted by $\mathcal{L}(\Theta; \text{RNN})$ in Fig 1) which represents how well the model predicts subjects' choices. In order to prevent the model from overfitting the data, we used early stopping [a commonly used regularisation method in deep learning; 30] and used cross-validation to assess the generalization abilities of the model to predict unseen data.

Baseline models. We compared the predictive accuracy of the RNN model with classical exemplars from the reinforcement learning (RL) family as well as a logistic regression model. The first baseline RL model was the *Q*-learning model (denoted by QL), in which subjects' choices are determined by learned action values [often called *Q* values; 31], which are updated based on the experience of reward [as used for example in 32]. The second baseline model was *Q*-learning with perseveration (denoted by QLP), which is similar to QL but has an extra parameter that allows for a tendency to stick with the same action for multiple trials (i.e., to perseverate), or sometimes to alternate between the actions [independently of reward effects; 4, 33]. As we show below, the accounts of subject choices provided by both QL and QLP were significantly worse than RNN, and so we developed a new baseline model that we called generalised *Q*-learning (denoted by GQL). This model extends QL and QLP models by learning multiple values for each action using different learning rates, and also by tracking the history of past actions at different time scales. The final baseline model we used was a logistic regression model (denoted by LIN) in which the probability of taking each action was determined by a linear combination of previous rewards, actions and their interactions [33, 34]. See section Computational models in Materials and methods for more details.

Task and subjects. The instrumental learning task (Fig 2) involved participants choosing between pressing a left (L action) or right (R action) button (self-paced responses) in order to earn food rewards (an M&M chocolate or a BBQ flavoured cracker). The task was divided into 12 different blocks each lasting for 40 seconds and separated by a 12-second inter-block interval. Within each block one of the actions was better than the other in terms of the probability of earning a reward but across blocks the action with the higher reward probability was varied, i.e., in some of the blocks the left action was better while in others the right action was better. The reward probability for the better action was 0.25, 0.125, or 0.08 and the probability of earning reward from the other action was always 0.05 (probabilities were fixed within each block); as such, there were six pairs of reward probabilities and each was repeated twice. 34 uni-polar depression (DEPRESSION), 33 bipolar (BIPOLAR) and 34 control (HEALTHY) participants (age, gender, IQ and education matched) completed the task. See Materials and methods for the details.

Performance in the task

Fig 3 shows the probability of selecting the best action (i.e., the action with the higher reward probability). Results are shown by subject (i.e., SUBJ) in the graph. The probability of selecting the better action was significantly higher than the other action in all groups (HEALTHY [η = 0.270, SE = 0.026, $p < 0.001$], DEPRESSION [$\eta = 0.149$, SE = 0.028, $p < 0.001$], BIPOLAR [$\eta = 0.119$,

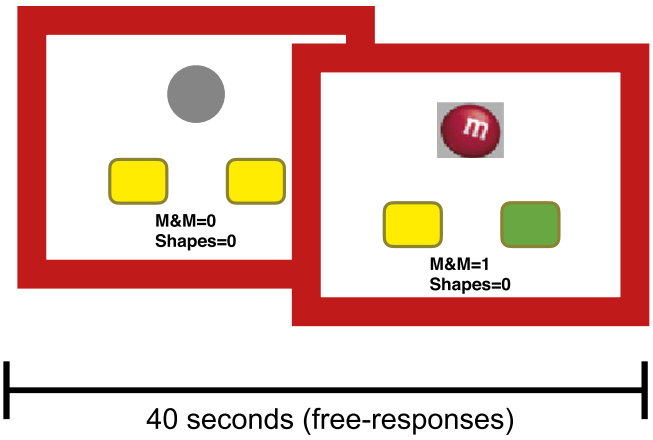

Fig 2. Structure of the decision-making task. Subjects had a choice between a left keypress (L) and a right keypress (R), shown by yellow rectangles. Before the choice, no indication was given as to which button was more likely to lead to reward. When the participant made a rewarded choice, the button chosen was highlighted (green) and a picture of the earned reward was presented for 500ms (M&M chocloate in this case). The task was divided into 12 different blocks each lasting for 40 seconds and separated by a 12-second inter-block interval. Within each block actions were self-paced and participants were free to complete as many trials as they could within the 40 second time limit. The probability of earning a reward from each action was varied between the blocks. See the text for more details about the probabilities of earning rewards from actions.

https://doi.org/10.1371/journal.pcbi.1006903.g002

SE = 0.021, $p < 0.001$]). Comparing HEALTHY and DEPRESSION groups revealed that the group x action interaction had a significant effect on the probability of selecting actions [$\eta = -0.120$, SE = 0.038, $p = 0.002$]. A similar effect was observed when comparing the HEALTHY and BIPOLAR groups [$\eta = -0.150$, SE = 0.034, $p < 0.001$]. In summary, these results indicate that all groups were able to direct their actions toward the better choice, however the DEPRESSION and BIPOLAR groups were less able to do so compared to the HEALTHY group.

Next, we trained three instances of a RNN using the data from each group and then froze the weights of the models and simulated them on-policy in the task (with the same reward probabilities and for the same number of trials that each subject completed). On-policy means that the models completed the task on their own by selecting the actions that they predicted a representative subject would take in each situation. The results of the simulations are shown in Fig 3 in the RNN column. Similar to the subjects' data, the probability of selecting the better action was significantly higher than the other action in all the three groups (HEALTHY [$\eta = 0.192$, SE = 0.011, $p < 0.001$], DEPRESSION [$\eta = 0.058$, SE = 0.014, $p < 0.001$], BIPOLAR [$\eta = 0.074$, SE = 0.011, $p < 0.001$]). Therefore, although the structure of RNN was initially unaware that the objective of the task was to collect rewards, its actions were directed toward the better key by following the strategy that it learned from the subjects' actions.

We also trained three instances of each baseline model using the data from each group, and simulated these on the task. Fig 3 shows the results of the simulations. As the graph shows, the LIN model was also able to direct its choices toward the best action by learning the effect of past choices and rewards on the next actions. A similar pattern was observed for GQL, QLP and QL models in the figure, which is not surprising as the structure of these models includes value representations which can be used for reward maximization. Thus, all of the models were consistent with the subjects' performance in terms of being able to find and select the best actions in the task. Further details about the models' parameters and training can be found in the Supporting information (estimated parameters for QL, QLP and GQL models are shown in S3, S4 and S5 Tables respectively; the negative log-likelihood for each model is reported in S6 Table. See S8 Table for the effect of initialisation of the network on the negative log-likelihood of the trained

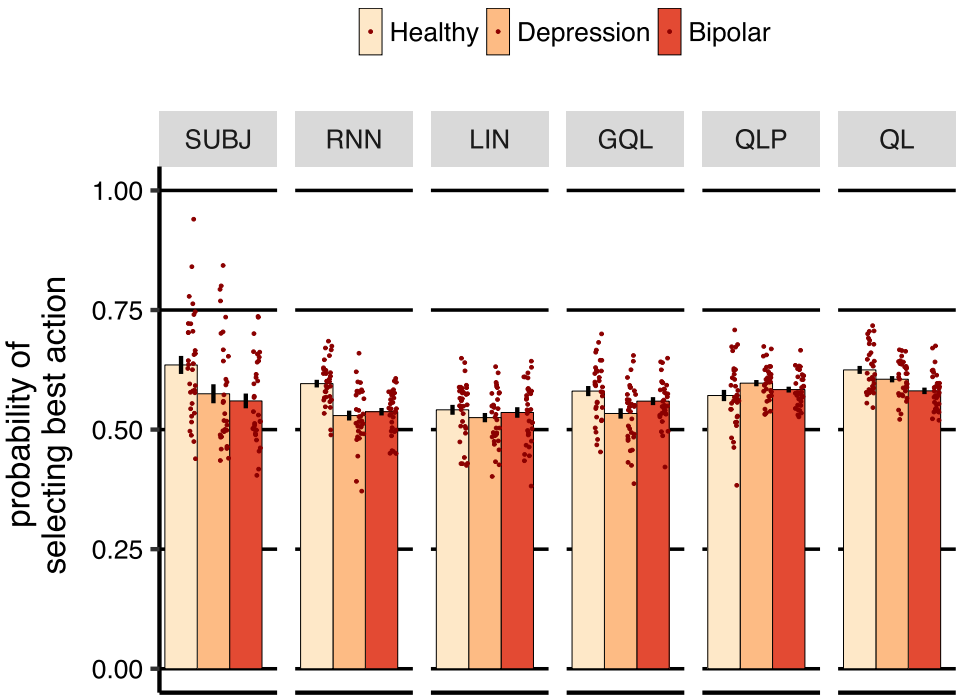

Fig 3. Probability of selecting the action with the higher reward probability (averaged over subjects). SUBJ refers to the data of the experimental subjects, whereas the remaining columns show simulations of the models trained on the task (on-policy simulations) with the same reward probabilities and for the same number of trials that each subject completed. Each dot represents a subject and error-bars represent 1 SEM.

https://doi.org/10.1371/journal.pcbi.1006903.g003

RNN. See S7 Table for the negative log-likelihood when a separate model was fitted to each subject in the case of baseline models. See S3 Text for the analysis of randomness of choices).

The immediate effect of reward on choice

The analyses in the previous section suggested that RNN was able to guide its actions toward the better choices, consistent with subjects' behavior. However, there are multiple strategies that the models could follow to achieve this, and here we aimed to establish whether the strategy used by the models was similar to that used by the subjects'. We started by investigating the immediate effect of reward on choice. Fig 4 shows the effect of earning a reward on the previous trial on the probability of staying on the same action in the next trial. For the subjects (SUBJ), earning a reward significantly *decreased* the probability of staying on the same action in the HEALTHY and DEPRESSION groups, but not in the BIPOLAR group (HEALTHY [$\eta = 0.112$, SE = 0.019, $p < 0.001$], DEPRESSION [$\eta = 0.111$, SE = 0.029, $p < 0.001$], BIPOLAR [$\eta = 0.030$, SE = 0.035, $p = 0.391$]). As the figure shows, the same pattern was observed in RNN (HEALTHY [$\eta = 0.082$, SE = 0.006, $p < 0.001$], DEPRESSION [$\eta = 0.089$, SE = 0.013, $p < 0.001$], BIPOLAR [$\eta = 0.001$, SE = 0.010, $p = 0.887$]), which shows that the strategy used by RNN was similar to the subjects' according to this analysis.

In contrast, stay probabilities were in the opposite directions in QL and QLP, i.e., the probability of staying on the same action was *higher* after earning reward (for the case of QLP; HEALTHY [$\eta = -0.028$, SE = 0.004, $p < 0.001$], DEPRESSION [$\eta = -0.039$, SE = 0.006, $p < 0.001$], BIPOLAR [$\eta = -0.054$, SE = 0.007, $p < 0.001$]), which differs from the subjects' data. This pattern was expected from baseline reinforcement-learning models, i.e., QL and QLP, because, in these models, earning reward increases the value of the taken action, which raises the probability of

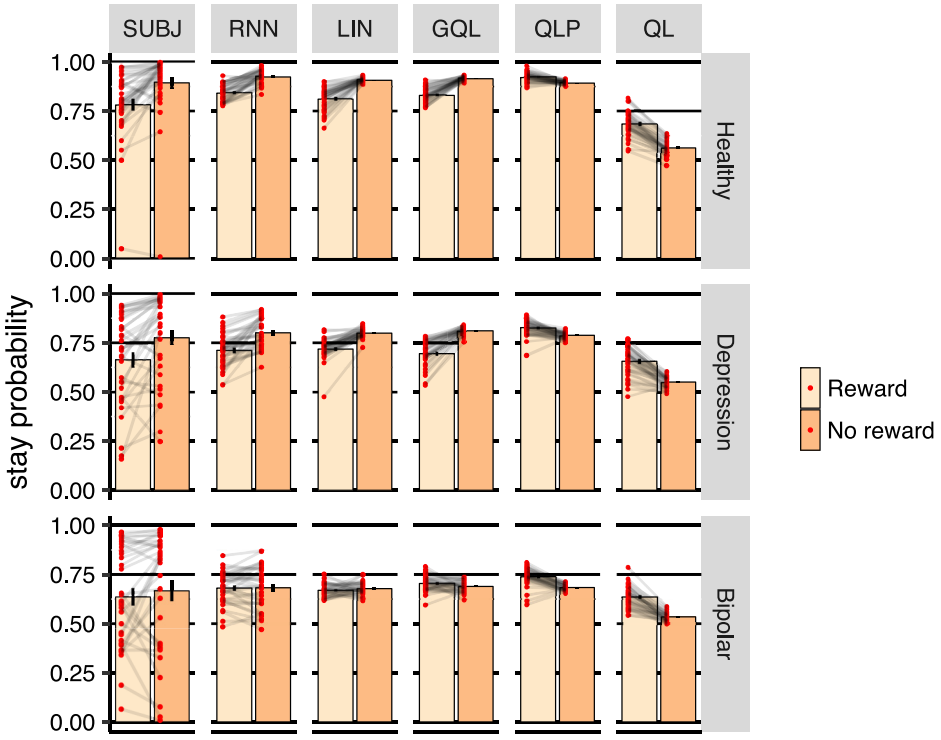

Fig 4. Probability of staying on the same action based on whether the previous trial was rewarded (reward) or not rewarded (no reward), averaged over subjects. SUBJ shows the data from subjects and results of columns are derived from on-policy simulations of various models on the task. Each dot represents a subject and error-bars represent 1SEM.

https://doi.org/10.1371/journal.pcbi.1006903.g004

choosing that action on the next trial. Indeed, this learning process is embedded in the parametric forms of QL and QLP models, and cannot be reversed no matter what values are assigned to the free-parameters of these models. Therefore, although QLP and QL were able to find the best action in the task, analyzing the immediate effect of reward showed that their learning processes differed from those used by the subjects'.

To address this limitation of QL and QLP, we designed GQL as a baseline model with more relaxed assumptions, in which action values could have an opposite effect on the probability of selecting actions, and so could generate a similar response pattern to the subjects', as shown in the figure. The LIN model was also able to produce a pattern similar to the empirical data. This is because in this model actions are determined by previous rewards (and actions) without making assumptions about the direction of the effects. Therefore, GQL and LIN appear to be able to model subjects's choices, at least in terms of the behavioral summaries presented here and in the previous section. Despite this, it remains an open question whether these models can represent all of the behavioural trends in the data, or whether there are some missing trends that were undetected in the summary statistics. In the next section we will answer this question by comparing the prediction capacity of GQL and LIN with RNN, as a model that has the capacity to capture all of the behavioural trends in the data.

Action prediction

Here our aim was to quantify how well the models predicted the actions chosen by the subjects. We used leave-one-out cross-validation for this purpose in which, at each round, one of the subjects was withheld and the model was trained using the remaining subjects; the trained

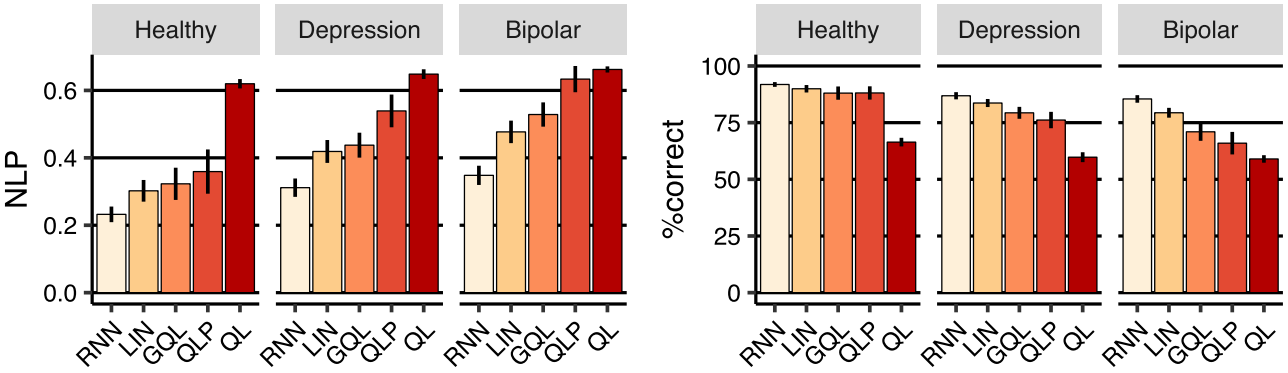

Fig 5. Cross-validation results. (Left-panel) NLP (negative log-probability) averaged across leave-one-out cross-validation folds. Lower values are better. **(Right-panel)** Percentage of actions predicted correctly averaged over cross-validation folds. Error-bars represent 1SEM.

https://doi.org/10.1371/journal.pcbi.1006903.g005

model was then used to make predictions about the withheld subject. The withheld subject was rotated in each group, yielding 34, 34 and 33 prediction accuracy measures in the HEALTHY, DEPRESSION, and BIPOLAR groups, respectively.

The results are reported in Fig 5. The left-panel of the figure shows prediction accuracy in terms of NLP (negative log-probability; averaged over leave one-out cross-validation folds; lower values are better) and the right-panel shows the percentage of actions predicted correctly ('%correct'; higher values are better). NLP roughly represents how well each model fits the choices of the withheld subject and so, unlike '%correct', takes the certainty of predictions into account. Therefore, we focus on NLP in this analysis. Firstly, LIN had the best NLP among the baseline models in all the three groups, although it was not statistically better than GQL. The GQL model was the second best among the baseline models and its advantage over QLP was statistically significant in the DEPRESSION and BIPOLAR groups (HEALTHY [$\eta = -0.036$, SE = 0.020, $p = 0.086$], DEPRESSION [$\eta = -0.101$, SE = 0.024, $p < 0.001$], BIPOLAR [$\eta = -0.105$, SE = 0.019, $p < 0.001$]). Secondly, RNN's NLP was even better than the best baseline model (LIN) across all groups (HEALTHY [$\eta = 0.069$, SE = 0.017, $p < 0.001$], DEPRESSION [$\eta = 0.107$, SE = 0.012, $p < 0.001$], BIPOLAR [$\eta = 0.128$, SE = 0.012, $p < 0.001$]), showing that RNN was able to predict subjects' choices better than the other models.

The fact that LIN and GQL were better than QL and QLP is not unexpected; we showed in the previous section that the predictions from QL and QLP were inconsistent with the trial-by-trial behaviour of the subjects. On the other hand, the fact that RNN is better than LIN and GQL shows that there are some behavioural trends that even LIN and GQL failed to capture, although they were consistent with subjects' choices according to the behavioural summary statistics. In the next sections, we use off-policy simulations of the models to uncover the additional behavioural trends that were captured by RNN.

Off-policy simulations

In an off-policy simulation, a model uses information about previous choices and rewards to make predictions about the next action. However, the actual next action used to simulate the model is not derived from these predictions and is derived in some other manner; notably, from human choices. In this way we can control the inputs the model receives and examine how they affect predictions. We were interested, in particular, in establishing how the predictions of the models were affected by the history of previous actions and rewards. As such, we designed a variety of inputs based on the behavioural statistics, fed them into the models, and recorded the predictions of each model in response to each input set (see S2 Text for more

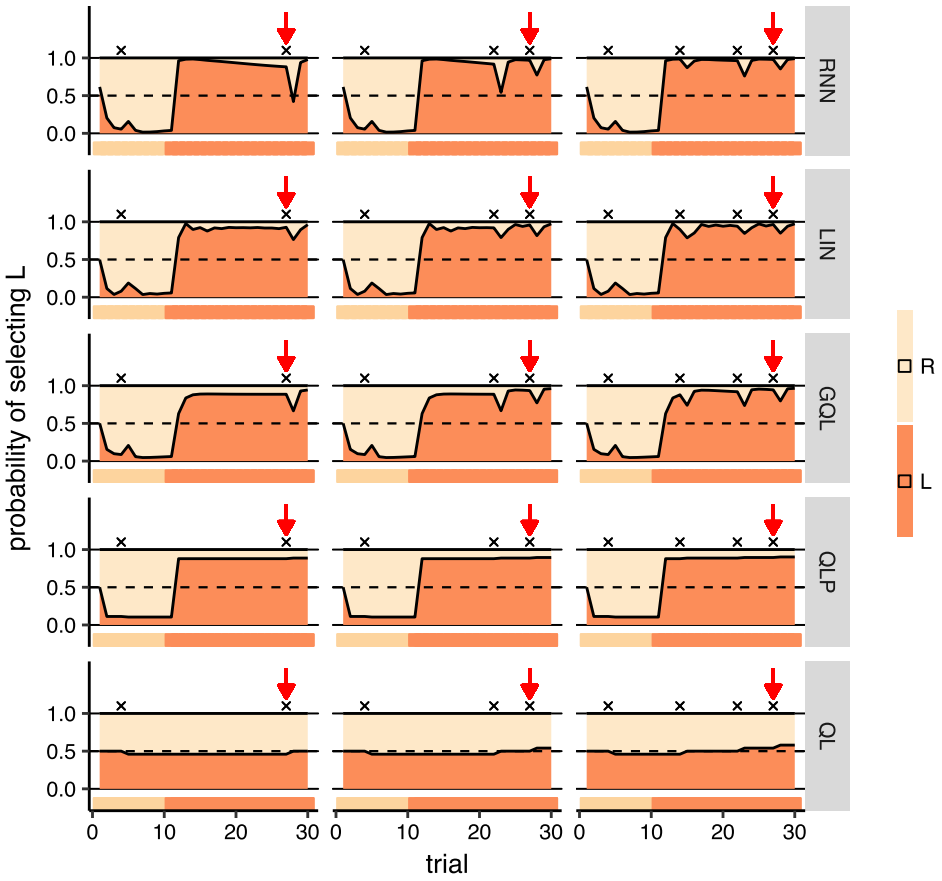

Fig 6. Off-policy simulations of all models for group HEALTHY. Each panel shows a simulation of 30 trials (horizontal axis), and the vertical axis shows the predictions of each model on each trial. The ribbon below each panel shows the action which was fed to the model on each trial. In the first 10 trials, the action that the model received was R and in the next 20 trials it was L. Rewards are shown by black crosses (x) on the graphs. Red arrows point to the same trial number in all the simulations and are shown to compare changes in the predictions in that trial between different simulations. The sequence of rewards and actions fed to the model are the same for the panels in each column, but they are different across the columns. See text for the interpretation of the graph.

https://doi.org/10.1371/journal.pcbi.1006903.g006

details on how simulation parameters were chosen). Simulations of the models are shown in each row of Fig 6 for the HEALTHY group, in which each panel shows a separate simulation across 30 trials (horizontal axis). For trials 1-10, the action that was fed to the model was R (right action), and for trials 11-30 it was L, i.e., left action (the action fed into the model at each trial is shown in the ribbons below each panel). The rewards associated with these trials varied among simulations (the columns) and are shown by black crosses (x) in the graphs.

The effect of reward on choice. Focusing on the RNN simulations in Fig 6, it can be observed that earning a reward (shown by black crosses) caused a 'dip' in the probability of staying with an action, which showed a tendency to switch to the other action. This is consistent with the observation made in Fig 4 that the probability of switching increases after reward. We saw a similar pattern in GQL, in which the contribution of action values to choices can be negative, i.e., higher values can lead to a lower probability of staying with an action (see S1 Text for more explanation). Similarly, in the LIN model the effect of reward on the probability of the next action can be negative, which allowed this model to produce the observed pattern. The pattern, however, is reversed in the QL and QLP models, i.e., the probability of choosing an action increased after a reward due to an increase in action value, which is again consistent

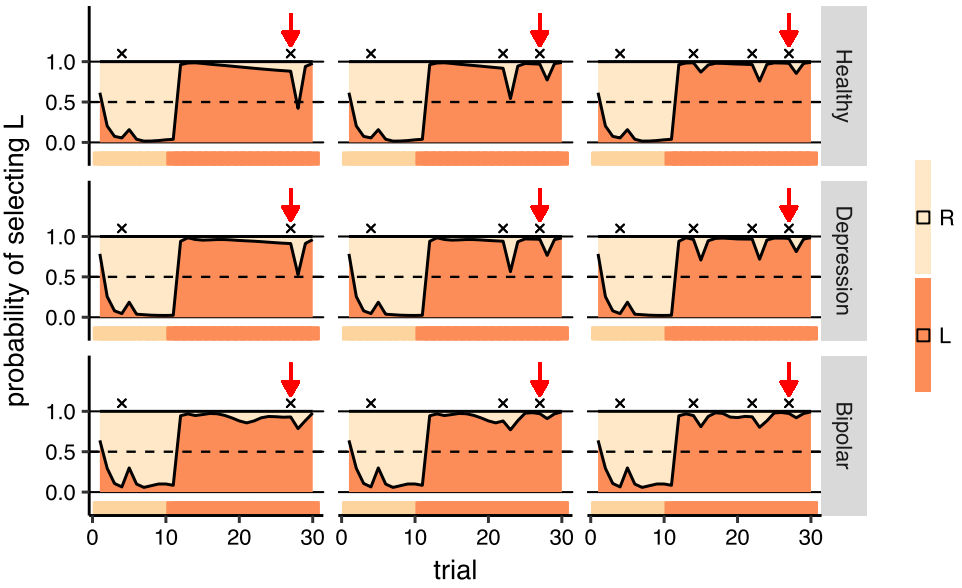

Fig 7. Off-policy simulations of RNN for all groups. Each panel shows a simulation of 30 trials (horizontal axis), and the vertical axis shows the predictions for each group on each trial. The ribbon below each panel shows the action which was fed to the model on each trial. In the first 10 trials, the action that the model received was R and in the next 20 trials it was L. Rewards are shown by black crosses (x) on the graphs, and the red arrows point to the same trial number in all the panels. See text for the interpretation of the graph. Note that the simulation conditions are the same as those shown in Fig 6, and the first row here (HEALTHY group) is the same as the first row shown in Fig 6 which is shown again for comparison with the other groups.

https://doi.org/10.1371/journal.pcbi.1006903.g007

with the observations in Fig 4. The effects are rather small in these two models (and may not be clear for the QLP model), which is likely because the effect of reward needs to be non-zero in order to enable the model to direct choices toward the best action in the long run. At the same time the effect is in the wrong direction compared to the actual data and therefore it needs to be kept small to minimize the discrepancy of predictions with the actual actions after earning rewards.

The next observation was the effect of the previous reward on the probability of switching after a reward. First we focused on the RNN model and on the trials shown by red arrows in Fig 6. The red arrows point to the same trial number, but the number of rewards earned prior to the trial differed. As the figure shows, the probability of switching after reward was lower in the right-panel compared to the left and middle panels. The only difference between simulations is that, in the right panel, two more rewards were earned before the red arrow. Therefore, the figure shows that although the probability of switching was higher after reward, it got smaller as the number of rewards previously earned by an action increased. Indeed, this strategy made subjects switch more often from the inferior action, because rewards were sparse on that action, and switch less from the best action, because it was more frequently rewarded. This reconciles the observations made in Figs 3 and 4 that, although more responses were made on the better action, the probability of switching after reward was higher. Fig 7 shows the same simulations using RNN for all groups. Comparing the predictions at the red arrows for the DEPRESSION and BIPOLAR groups, we observed a pattern similar to the HEALTHY group, although the differences were smaller in the BIPOLAR group (see S11 Fig for the effect of the initialisation of the model).

The above observations are consistent with the pattern of choices in the empirical data shown in Fig 8-left panel, which depicts the probability of staying with an action after earning

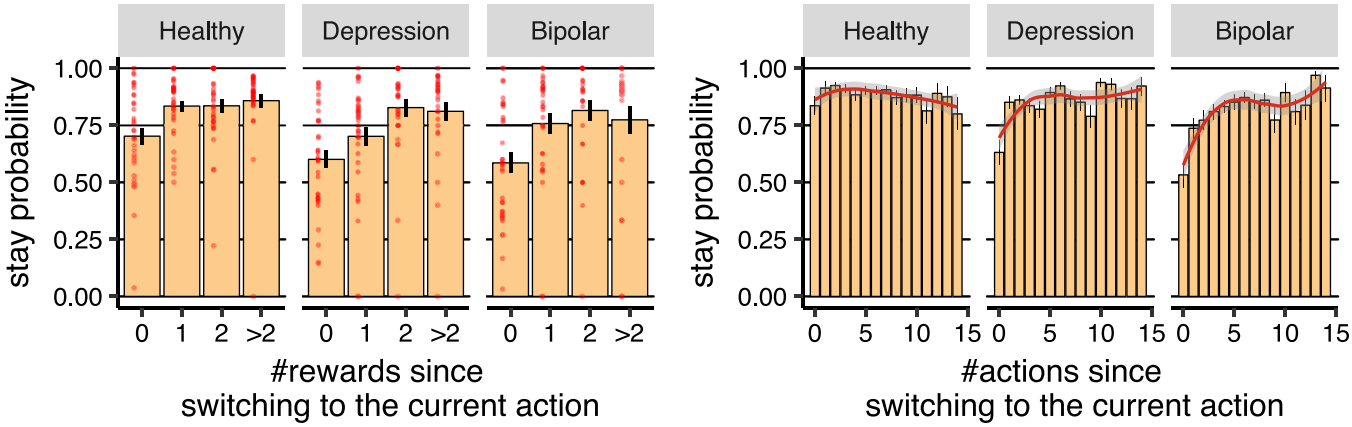

Fig 8. The effect of the history of previous rewards and actions on the future choices of the subjects. (Left-panel) The probability of staying with an action after earning reward as a function of the number of rewards earned since switching to the current action (averaged over subjects). Each red dot represents the data for one subject. **(Right-panel)** The probability of staying with an action as a function of the number of actions taken since switching to the current action. The red line was obtained using Loess regression (Local Regression), which is a non-parametric regression approach. The grey area around the red line represents the 95% confidence interval. Error-bars represent 1SEM.

https://doi.org/10.1371/journal.pcbi.1006903.g008

reward as a function of how many rewards were earned after switching to the action (a similar graph using on-policy simulation of RNN is shown in S13 Fig). In all three groups, the probability of staying with an action (after earning a reward) was significantly higher when more than two rewards were earned previously (>2) compared to when no reward was earned (HEALTHY [$\eta = 0.148$, SE = 0.037, $p < 0.001$], DEPRESSION [$\eta = 0.188$, SE = 0.045, $p < 0.001$], BIPOLAR [$\eta = 0.150$, SE = 0.056, $p = 0.012$]), which is consistent with the behaviour of the RNN.

As shown in Fig 6, the GQL model produced a pattern similar to RNN largely because this model tracks multiple values for each action, which allows this model to produce the 'dip's after reward, with a magnitude that is sensitive to the number of past rewards (see S1 Text for details). As such, it is not surprising that GQL was consistent with the subjects' choices with respect to the effects of immediate reward. Similarly, the LIN model was able to produce this pattern, which is again expected as in this model the predicted probabilities are based on rewards earned several trials back (up to 18 trials back), which allows this model to learn about the effect of distant rewards on current actions. In this case, the rewards earned proximal to the action will have a negative effect on selecting the same action—generating the observed 'dip' in the probabilities—and distant rewards will have a positive effect on the probabilities of staying on the same action, making the size of the 'dip' sensitive to the number of rewards earned in the past (see S6, S7, S8, S9 Figs for the corresponding results for all of the groups based on the LIN, GQL, QLP and QL models respectively).

The effect of repeating an action on choices. Next, we looked at the effect of the history of actions on choices. Focusing on the RNN model in Fig 6, we can see that, in the first 10 trials, the predicted probability of taking R was higher than L; this then reversed over the next 20 trials. This implies that perseveration (i.e., sticking with the previously taken action) was an element of action selection. This is consistent with the fact that the QLP model (which has a parameter for perseveration) performed better than the QL model in the cross-validation statistics (see Fig 5); and, indeed, Fig 6 shows QL's inability to reflect this characteristic. Note that it can be seen in Fig 3 that the probability of staying with an action was above 50% irrespective of whether a reward was earned on the previous trial or not. This does not, however, provide evidence for perseveration because the trials were not statistically independent. For example, in late training trials a subject might have discovered which action returns more reward on

average and, therefore, stayed with that action irrespective of reward and so without necessarily relying on perseveration.

Focusing on RNN simulations in the left-panel of Fig 6, we observed that, after switching to action L (after trial 10), the probability of staying with that action gradually decreased; i.e., although there was a high chance the next action would be similar to the previous action, subjects developed a tendency to switch the longer they stayed with an action. To compare this pattern with the empirical data, we calculated the probability of staying with an action as a function of how many times the action had been taken since switching (Fig 8:right-panel; similar graphs for RNN, LIN and GQL on-policy simulations are shown in S13, S14 and S15 Figs respectively). As the figure shows, for the HEALTHY group, the chance of staying with an action decreased as the action was repeated [$\eta = -0.005$, SE = 0.001, $p = 0.006$], which is consistent with the behaviour of RNN. With regard to the baseline models, going back to Fig 6, we did not see a similar pattern, although in GQL there was a small decrement in the probability of staying with an action after earning the first reward.

Symmetric oscillations between actions. Next, we focussed on the RNN simulations in Fig 7 in DEPRESSION and BIPOLAR groups for which the gap between prediction accuracy of baseline models and RNN was largest. As can bee seen in the left-panels, after switching to action L (after trial 10), the probability of staying with that action gradually decreased in the DEPRESSION group whereas, for the BIPOLAR group, there was a 'dip' around 10 trials after the switching to action L (i.e., around trial 20), and then the policy became flat. With reference to the empirical data, as shown in Fig 8:right-panel, for the DEPRESSION and BIPOLAR groups, the probability of staying with an action immediately after switching to that action was around 50%—60% (shown by the bar at $x = 0$ in Fig 8:right-panel), i.e., there was a 40%–50% chance that the subject immediately switched back to the previous action. Based on this we expected to see a 'dip' in the simulations of the DEPRESSION and BIPOLAR groups in Fig 7 just after the switch to action L. This was not the case, pointing to an inconsistency between model predictions and the empirical data.

However, Fig 7 is based on particular, artificial sequences of actions and rewards. To look more closely at the above effect, we defined a *run* of actions as a sequence of presses on a specifed button without switching to the other button. For example, if the executed actions were L, R, R, L, then the length of the first run was 1 (L), the length of the second run was 2 (R, R), and the length of the third run was 1 (L). Fig 9 shows the relationship between consecutive run length, i.e., the length of the current run of actions, as a function of the length of the previous run of actions in the empirical data. The graph show the empirical data (shown by SUBJ) and on-policy model simulations. The dashed line in the figure indicates the points at which the current run length was the same as the previous run length. Being close to this line implies that subjects were performing symmetrical oscillations between the two actions, i.e., going back and forth between the two actions while performing an equal number of presses on each button. The data for subjects (shown in SUBJ column) shows that, in the BIPOLAR group, and to some extent in the DEPRESSION group, a short run triggered a subsequent run of similar brevity (see S3, S4, and S5 Figs for raw empirical data). This implies that if a subject performed a run of length 1 that would initiate a sequence of oscillations between the two actions, keeping the stay probabilities low during short runs, consistent with what was seen at $x = 0$ in Fig 8:right-panel. This effect was not seen in the simulations shown in Fig 7, because the length of the previous run before switching to action L was 10 (there were 10 R actions), and therefore we should not expect the next run to be of length 1, nor should we have actually expected to see a 'dip' in policy just after the first switch.

As shown in S12 Fig, the modal length of runs in the DEPRESSION, and BIPOLAR groups was 1 (around 17%, 37%, and 45% of runs were of length 1 in the HEALTHY, DEPRESSION, and BIPOLAR groups respectively). Given this, and the specific pattern of oscillations in the DEPRESSION and

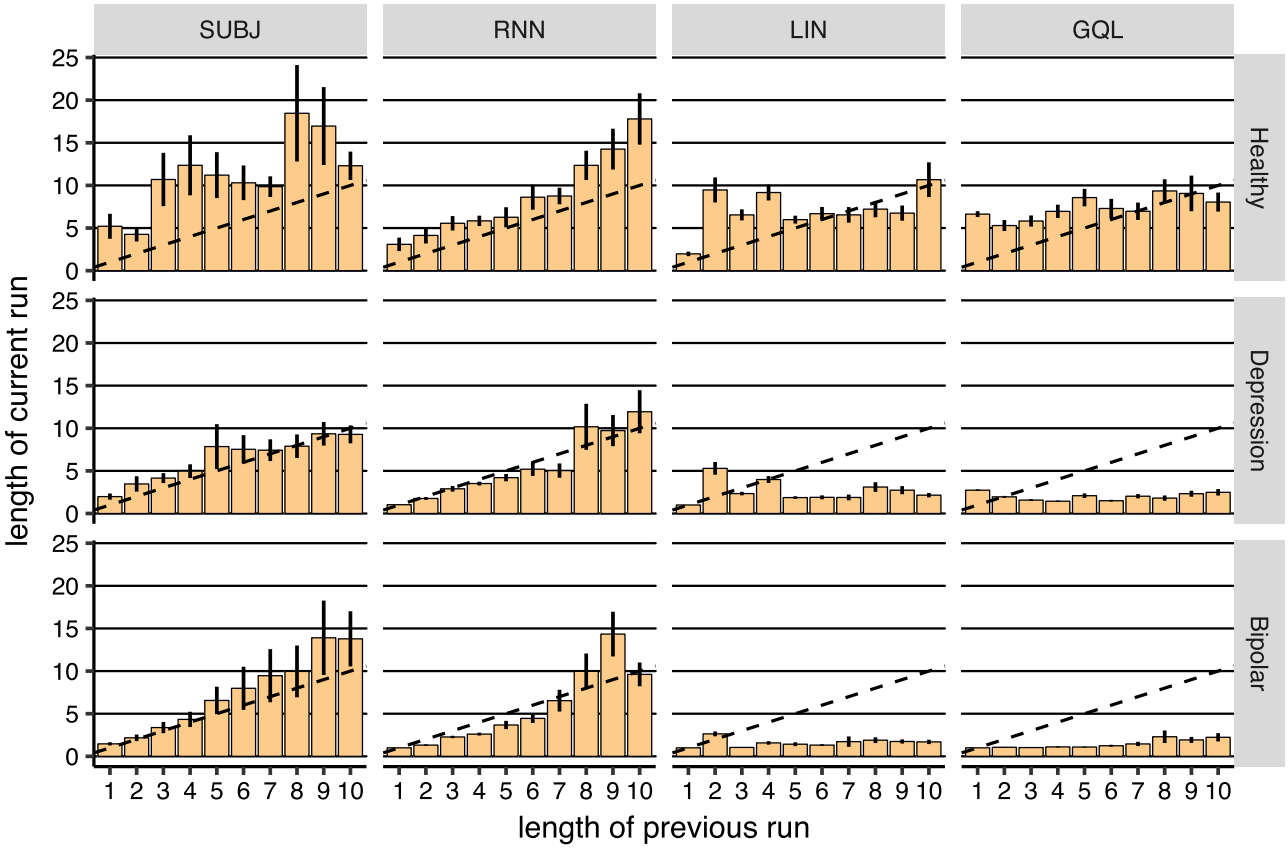

Fig 9. The median number of actions executed sequentially before switching to another action (run of actions) as a function of the length of the previous run of actions (averaged over subjects). The dotted line shows the points at which the length of the previous and the current run of actions were the same. Note that the median was used instead of the average to illustrate the most common 'current run length', instead of the average run length for each subject. The results for actual data are shown in SUBJ column, and the remaining columns show the results using the on-policy simulations of the models in the task. Error-bars represent 1SEM.

https://doi.org/10.1371/journal.pcbi.1006903.g009

BIPOLAR groups, our next question was whether, in the models, a run of length 1 triggered oscillations similar to those observed in the empirical data. We used a combination of off-policy and on-policy model simulations to answer this question; i.e., during the off-policy phase we forced the model to make an oscillation between the two actions, and then allowed the model to select between actions. We expected, in the HEALTHY group, that the model would converge on one action, whereas, in the DEPRESSION and BIPOLAR groups, we expected the initial oscillations to trigger further switches. Simulations are presented in Fig 10, which shows that the sequence of actions fed to the model for the first 9 (off-policy) trials was:

R, R, R, R, R, R, L, R, L,

in which there were two oscillations at the tail of the sequence (R, L, R, L,). The rest of the actions (trials 10-20) were selected based on which action the model assigned the highest probability. Note that in on-policy simulations, actions were typically selected probabilistically according to the probabilities that a model assigned to each action. However, in the on-policy simulations presented in this section, in order to get consistent results across simulations actions were *not* selected probabilistically but were chosen based on which action achieved the highest prediction probability. As the simulation shows, at the beginning, the probability the model assigned to action R was high, but after feeding in the oscillations, the model predicted

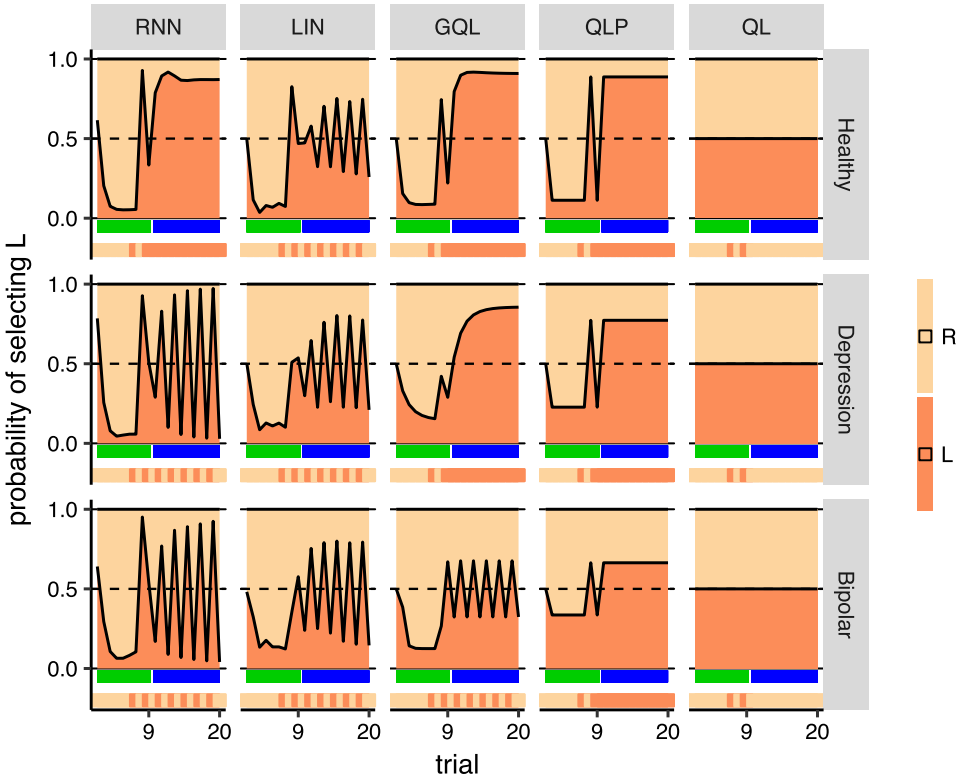

Fig 10. Mixed off-policy and on-policy simulations of the models. Each panel shows a simulation of 20 trials for which the first nine trials were off-policy and the subsequent trials were on-policy, during which the action with the highest probability was selected. Trials marked with green ribbons were off-policy (actions were fed to the model), whereas the trials marked with blue ribbons were on-policy (actions were selected by the model). The ribbon below each panel shows the actions that were fed to the model (for the first 9 trials), and the actions that were selected by the model (on the subsequent trials). During off-policy trials, the sequence of actions that was fed to the model was R, R, R, R, R, R, L, R, L. See the text for interpretation.

https://doi.org/10.1371/journal.pcbi.1006903.g010

that the future actions would oscillate in the DEPRESSION and BIPOLAR groups, but not in the HEALTHY group, consistent with what we expected to observe.

Therefore, RNN was able to produce symmetrical oscillations and its behaviour was consistent with the subjects' actions. As Fig 10 shows, besides RNN, the LIN and GQL models were also able to produce length 1 oscillations to some extent (as shown for the BIPOLAR group), which could explain why the prediction accuracy achieved by these models was significantly better than QLP in the BIPOLAR and DEPRESSION groups (Fig 5) in which length 1 oscillations were more common (see S2 Text for more details). However, as shown in Fig 9, LIN and GQL both failed to produce oscillations of longer lengths, whereas RNN was able to do so (Fig 9 RNN column). GQL failed to produce the oscillations even if we increased the capacity of GQL to track 10 different values for each action (see S16 Fig). Similarly, as the simulations show, the LIN model was not able to produce these oscillations; such oscillations will likely require higher order interaction terms and, although these could be added to the LIN model in principal, in practise they will significantly complicate its transparent interpretation. This failing of the LIN and GQL models is particularly problematic in the DEPRESSION and HEALTHY groups, because these two groups tended to match the length of consecutive runs of actions. This could partly account for why the cross-validation statistics associated with RNN were significantly better than LIN and GQL for the DEPRESSION and BIPOLAR groups.

Table 1. Prediction of diagnostic labels using RNN. The number of subjects in each true- and predicted-label. The numbers inside the parentheses are the percentage of subjects relative to the total number of subjects in each diagnostic group.

		predicted labels		
		HEALTHY	DEPRESSION	BIPOLAR
true labels	HEALTHY	22 (64%)	8 (23%)	4 (11%)
	DEPRESSION	13 (38%)	16 (47%)	5 (14%)
	BIPOLAR	9 (27%)	9 (27%)	15 (45%)

https://doi.org/10.1371/journal.pcbi.1006903.t001

Summary of off-policy simulations. Firstly, we found that a RNN model was able to capture the immediate effect of rewards on actions (i.e., the 'dip' after rewards), as well as the effect of previous rewards on choices. GQL and LIN had a similar ability, which enabled them to reproduce the behavioural summary statistics shown in Figs 3 and 4. Baseline reinforcement-learning models (QLP and QL) failed to capture either trend. Secondly, RNN was able to capture how choices change as an action is chosen repeatedly and sequentially, and also the symmetrical oscillations between actions, neither of which could be detected by any of the baseline models.

Diagnostic label prediction

In the previous sections we showed that there are several behavioural trends that baseline models failed to capture. Here we asked whether capturing such behavioural trends in this task is necessary to predict the diagnostic labels of the subjects. We used the leave-one-out cross-validation method based on which, in each run, one of the subjects in each group was withheld, and a RNN model was fitted to the rest of the group. This model, along with the versions of the same model fitted to all of the subjects in each of the other two groups, was used to predict the diagnostic label for the withheld subject. This prediction was based on which of the three models provided the best fit (lowest NLP) for that subject. As an example, assume that we are interested to predict the diagnostic label for a certain subject—say subject #10—in the DEPRESSION group. We first trained a model using the data of all other subjects in the DEPRESSION group, and we also trained two other models using the data of HEALTHY and BIPOLAR groups. Then, we evaluated which of these three models can better predict the actions of subject #10 in terms of NLP to predict the diagnostic label of the subject.

The results are reported in Table 1. Baseline random performance was near 33%. As the table shows, the highest performance was achieved for the HEALTHY group of which 64% of subjects were classified correctly. A binomial test indicated that the proportion of correctly classified subjects in the HEALTHY group was significantly different than the expected proportion of 0.336 based on random classification ($p < 0.001$ two-sided). On the other hand, in the DEPRESSION group a significant portion of subjects were classified as HEALTHY. A binomial test did not indicate that the proportion of correctly classified subjects in the DEPRESSION and BIPOLAR groups was significantly different than the expected proportion of 0.336 and 0.326, respectively, based on random classification ($p > 0.1$ two-sided).

The overall correct classification rate of the model was 52%, whereas LIN achieved 46% accuracy (S1 Table) and GQL achieved 50% accuracy (S2 Table). We conclude that although LIN and GQL were unable to accurately characterize behavioural trends in the data, the group differences that were captured by LIN and GQL appeared sufficient to guide diagnostic label predictions.

Discussion

We used a recurrent neural network to provide a framework for learning a computational model that can characterize human learning processes in decision-making tasks. Unlike

previous work, the current approach makes minimal assumptions about these learning processes; we showed that this agnosticism is important in developing an appropriate explanation of the data. In particular, the RNN model was able to encode the melange of processes that subjects appeared to use to select actions; it was also able to capture differences between the psychiatric groups. These processes were largely inconsistent with conventional and tailored *Q*-learning models, and were also hidden in the overall performance of subjects on the task. This provided a clear example of how the currently proposed framework can outperform previous approaches.

In general, as we were able to show, this new approach improves upon previous methods from four standpoints: First, it provides a model that was able to predict subjects' choices without requiring manual engineering; and to do so more accurately than baseline models on test data. Second, the framework contributes to computational modelling by providing a baseline for predictive accuracy; i.e., to the extent that other candidate models failed to generate the performance of RNN models, important and accessible behavioural trends would have been missed in the model structure. This is particularly important because the natural randomness of human choice in many scenarios makes it unclear whether the model at hand (e.g., a *Q*-learning model) has reached a limit as to how well those choices can be predicted, or whether it requires further improvements. Without other recourse, conventional treatments tend to relegate model mis-fit to irreducible randomness in choice. Third, based on the framework, a trained model can be regarded as representative of a group's behaviour, which can then be interrogated in control conditions using off-policy simulations to gain insights into the learning processes behind subject's choices. Finally, the framework can be used to predict the diagnostic labels of the subjects.

It might be possible to design different variants of *Q*-learning models (e.g., based on the analysis presented above) and obtain more competitive prediction accuracy. For example, although it is non-trivial, it is possible to design a new variant of GQL able to track oscillatory behaviour such as that described here. Our aim was not to rule out this possibility, but rather to show that the framework can automatically extract learning features from subjects' actions using learning to learn principles without requiring feature engineering in the models. This is even when those features were initially invisible in task performance metrics.

Our approach inherits these benefits from the field of neural networks [35], in which feature engineering has been significantly simplified across various domains [see also for example 36, for recent developments using probabilitic programming]. However, our approach also inherits the black-box nature of these neural networks, i.e., the lack of an interpretable working mechanism. This might not be an issue in some applications, such as the ones mentioned above; however, this needs to be addressed in other applications in which the aim of the study is actually to obtaining an interpretable working mechanism. Nevertheless, we were able to show that running controlled experiments on the model using off-policy simulations can provide significant insights into the processes that mediate subjects' choices. An alternative method for interpreting the model is using gradients [37], which will be considered in future work. Interpreting neural networks is an active area of research in machine learning [e.g., 38], and the approach proposed here will benefit from further developments in this area.

In particular, although we found that off-policy simulations of the model could be used to gain insights into the model's working mechanism, off-policy simulations need to be designed manually to determine inputs to the model. Here, we designed the initial off-policy simulations based on the specific questions and hypotheses that we were interested in testing and using overall behavioural statistics (Fig 6; S2 Text). However, an important aspect of the behavioural process, i.e., the tendency of subjects to oscillate between actions, was not visible in those simulations and, because of this, we had to design another set of inputs to investigate

these oscillations (Fig 10). This shows that the choice of off-policy simulation can affect the interpretation of the model's working mechanism. As such, although RNN can be trained automatically and without intuition into the behavioural processes behind actions [e.g., 39], designing off-policy simulations is not automated and requires manual hypothesis generation. Automating this process will require a method that generates representative inputs (and network outputs) that are clearly able to discriminate the differences between the psychiatric groups. The existence of adversarial examples in neural networks [40] suggests that this will not be as simple as using the networks to search explicitly for those input sequences that are the most discriminative—representativeness is also critical.

Recurrent neural networks have previously been used to study reward-related decision-making [13, 14], perceptual decision-making, performance in cognitive tasks, working-memory [15, 16, 17, 18, 19, 20], motor patterns, motor reach and timing [21, 22, 23, 24]. Typically, in these studies, an RNN is itself trained to perform the task. This is different from the current study in which the aim of training was to generate behaviour similar to the subjects', even if that were to lead to poor performance on the task. One exception is the study of [21] in which a network was trained to generate outputs similar to electromyographic (EMG) signals recorded in behaving animals during a motor reach task. Interestingly, that study found that, even though the model was trained based purely on EMG signals, the internal activity of the model resembled the neural responses recorded from the subjects' motor cortex. Indeed, we have recently used a similar approach to investigate whether brain activity during decision-making is related to network activity [37].

The fact that RNN performance was better than the baseline models can be attributed to two factors. Firstly, recurrent neural networks can potentially track a long history of previous events (such as rewards and actions) in order to predict the next actions. Evidence shows that humans tend to find patterns in the history of previous events (e.g., repetitions, alterations)—even if the events are generated randomly—and subsequently use those patterns to guide their choices and therefore it is important that a model be able to represent such inter-dependencies between current actions and past events [e.g., sequential effects; 41, 42, 43, 44, 45, 46, 47]. Secondly, such past influences might have a non-linear effect on current choices, and therefore it is important that the model be able to track higher-order statistics [48, 49] and non-linear effects. For example in the current study a linear logistic regression model (LIN) was unable to reproduce the symmetrical oscillations between the actions whereas RNN could, which shows that non-linear dynamics are necessary to explain the current data.

With regard to predicting subjects' diagnostic labels, it was perhaps not surprising to find that the model was unable to achieve a high level of classification accuracy. This is because there is a high level of heterogeneity in patients with the same diagnostic label. Heterogeneity, which is well understood in the wide variation in treatments and treatment outcomes in disorders like depression [e.g., 50], is likely also to be reflected in the differing learning and choice abilities of the subjects.

Given a set of models, different (approximate) Bayesian methods can be used for comparing different candidate models in order to find the model that has generated the data (or has the highest probability of being the model that has generated the data). This comparison can be achieved for example by calculating model-evidence, Bayes factors, exceedance probabilities [51], or using hierarchical Bayesian model comparison [52]. In some other settings, however, the aim is not just to compare a set of models, but to develop new models or to improve them in order to achieve a high out-of-sample prediction accuracy. A natural way to assess such prediction accuracy is to use cross-validation, that we used jointly with early stopping [30] to prevent the RNN from overfitting the data. Indeed, it has been suggested that, from a Bayesian perspective, the other quantities such as Akaike information criterion [AIC; 53], Deviance

information criterion [DIC; 54, 55], and Watanabe-Akaike information criterion [WAIC; 56] can be viewed as approximations to different forms of cross-validation [57], which was directly calculated in the current study.

In the model fitting procedure used here, a single model was fitted to all of the subjects in each group, despite possible individual differences within a group. This was partly because we were interested in obtaining a single parameter set for making predictions for the subject withheld in the leave-one-out cross-validation experiments. Even if a mixed-effect model was fitted to the data, a summary of group statistics will be required to make predictions about a new subject. In other applications, one might be interested in estimating parameters for each individual (either network weights or the parameters of the reinforcement-learning models); in this respect using a hierarchical model fitting procedure would be a more appropriate approach, something that has been used previously for reinforcement-learning models [e.g., 4] and would be an interesting future step for RNN models.

Along the same lines, due to its rich set of parameters, a single RNN model might be able to learn about and detect individual differences (e.g., differences in the learning-rates of subjects) at an early stage of the task, and then use this information to make predictions about performance on later trials. For example, during the training phase (learning how humans learn), the model might learn that subjects have either a very high or a very low learning-rate. Then, when being evaluated in the actual learning task, the model can use observations from subjects' choices on early trials to determine whether the learning-rate for that specific subject is high or low, and then utilise that information to make more accurate predictions in latter trials. Determining individual-specific traits in early trials of the task is presumably *not* part of the computational process occurring in the subject's brain during the task, and is occurring in the model merely to make more accurate predictions. To the extent that the network learns such higher order structure, it is appealing, though difficult, to extract information about such heterogeneity from the recurrent state of the RNN. Of course, this implies that the (implicit) inferences that the RNN makes about the type of subject might be confounded with the (implicit) inferences that the RNN makes about the actual choices—thus it is a model that makes predictions about subjects' choices using mechanisms that may not necessarily be competent computational models of the way that the subjects themselves make those choices.

Materials and methods

Ethics statement

The study was approved by the University of Sydney ethics committee (HREC #12812). Participants gave informed consent prior to participation in the study.

Participants

34 uni-polar depression (DEPRESSION), 33 bipolar (BIPOLAR) and 34 control (HEALTHY) participants (age, gender, IQ and education matched) were recruited from outpatient mental health clinics at the Brain and Mind Research Institute, Sydney, and the surrounding community. Participants were aged between 16 and 33 years. Exclusion criteria for both clinical and control groups were history of neurological disease (e.g. head trauma, epilepsy), medical illness known to impact cognitive and brain function (e.g. cancer), intellectual and/or developmental disability and insufficient English for neuropsychological assessment. Controls were screened for psychopathology by a research psychologist via clinical interview. Patients were tested under 'treatment-as-usual' conditions, and at the time of assessment, 77% of depressed and 85% of

Table 2. Demographic and clinical characteristics of participants. Means (SD). HDRS: Hamilton Depression Rating Scale; YMRS: Young Mania Rating Scale; SOFAS: Social and Occupational Functioning Scale; a: DEPRESSION greater than HEALTHY and BIPOLAR, $p < 0.05$. b: BIPOLAR greater than HEALTHY, $p < 0.05$. c: HEALTHY greater than DEPRESSION and BIPOLAR, $p < 0.05$.

	HEALTHY (n = 34)	DEPRESSION (n = 34)	BIPOLAR (n = 33)
Demographics			
Gender (M:F)	15:19	15:19	9:24
Age in years	23.6 (4.3)	21.6 (2.5)	23.1 (4.4)
Predicted IQ	107.3 (7.5)	105.5 (7.9)	106.0 (7.4)
Eduation	14.3 (3.0)	13.3(1.9)	13.3 (2.4)
Symptoms and History			
Age of onset (years)	-	14.4 (3.8)	15.9 (4.7)
Duration of illness (years)	-	7.7 (4.3)	6.4 (3.3)
HDRS	1.5(2.0)	14.1 (7.2)[a]	8.9 (6.5)
YMRS	0.1 (0.4)	2.5 (5.4)	4.6 (5.8)[b]
SOFAS	91.0 (3.5)[c]	63.8 (9.2)	65.7 (13.7)
Medication			
Medicated	-	77%	85%
Anti-depressants	-	71%	41%
Mood stabilizers/Anti-convulsants	-	9%	73%
Lithium	-	0%	18%
Anti-psychotics	-	18%	33%
Anxiolytics	-	0%	3%
Motivation measures			
Hunger	6.5 (1.7)	6.0 (2.1)	6.0 (2.4)
Reward Pleasantness	3.1 (1.3)	2.0 (2.0)	2.6 (2.0)

Duration of illness indicates time since patient first experienced mental health problems, not time since diagnosis.

https://doi.org/10.1371/journal.pcbi.1006903.t002

bipolar patients were taking medications (see Table 2 for breakdown of medication use). The study was approved by the University of Sydney ethics committee. Participants gave informed consent prior to participation in the study.

Demographics and clinical characteristics of the sample are presented in Table 2. Levene's test indicated unequal variances for the HDRS [Hamilton Depression Rating Scale; 58], YMRS [Young Mania Rating Scale; 59], SOFAS [Social and Occupational Functional Scale; 60] and age, thus Welch's statistic was used for these variables. A one-way ANOVA revealed no differences between groups in age [$F(2, 98) = 2.48$, $p = 0.09$], education [$F(2, 98) = 1.76$, $p = 0.18$], IQ [$F(2, 94) = 0.47$, $p = 0.62$] or gender ($\chi^2 = 2.66$, $p = 0.27$). There were differences in HDRS [$F(2, 49.21) = 64.21$, $p < 0.001$], YMRS [$F(2, 43.71) = 12.57$, $p < 0.001$], and SOFAS [$F(2, 41.61) = 169.66$, $p < 0.001$]. Bonferroni post-hoc comparisons revealed higher depression scores in DEPRESSION group compared to BIPOLAR and HEALTHY groups, and higher depression in BIPOLAR group compared to HEALTHY group. Mania scores were significantly higher in the BIPOLAR group compared to the HEALTHY group. Both patient groups had significantly lower SOFAS scores compared to the HEALTHY group, but did not differ from one another. Age of mental illness onset was younger in the DEPRESSION group compared to the BIPOLAR group [$t(56) = -2.14$, $p = 0.04$], however duration of illness did not differ significantly between groups [$t(56) = 1.25$, $p = 0.22$]. There were no differences between groups in pre-test hunger [$F(2, 79) = 0.54$, $p = 0.59$] or average snack rating [$F(2, 79) = 2.53$, $p = 0.09$].

Task

The instrumental learning task (Fig 2) involved participants choosing between pressing a left or right button in order to earn food rewards (an M&M chocolate or a BBQ flavoured cracker). We refer to these two key presses as L and R for left and right button presses respectively. Fourteen HEALTHY participants (41.2% of the group) and 13 BIPOLAR participants (36.7% of the group) completed the task in an fMRI setting, using a 2 button Lumina response box. The remaining HEALTHY and BIPOLAR participants, and all DEPRESSION participants, completed the task on a computer with a keyboard, where the "Z" and "?" keys were designated L and R. Although the performance of subjects was higher overall in the fMRI setting [$\eta = 0.050$, SE = 0.024, $p = 0.041$], the place in which the task was completed had no significant effect on how choices adjusted on a trial-by-trial basis, either on the probability of staying with the same action after earning a reward [$\eta = 0.041$, SE = 0.054, $p = 0.45$], or after no reward [$\eta = 0.030$, SE = 0.062, $p = 0.627$]), and, therefore, the data were combined.

During each block, one action was always associated with a higher probability of reward than the other. The best action was varied across blocks, and the probabilities varied between 0.25, 0.125, and 0.08. The probability of reward on the other action always remained at 0.05. Therefore, there were six pairs of reward probabilities and each was repeated twice. Participants were instructed to earn as many points as possible, as they would be given the concomitant number of M&Ms or BBQ flavoured crackers at the end of the session. After a non-rewarded response, a grey circle appeared in the centre of the screen for 250ms, whereas after a rewarded response the key turned green and an image of the food reward earned appeared in the centre of the screen for 500ms. A tally of accumulated winnings remained on the bottom of the screen for the duration of the task. The task began with a 0.25 contingency practice block and a pleasantness rating for each food outcome (-5 to +5). Responding was self-paced during the 12 blocks of training, each 40-s in length. On average participants completed 109.45, 114.91, 102.79 trials per block in HEALTHY, DEPRESSION, and BIPOLAR groups respectively (see S9 Table for the average number of trials completed in each reward probability condition in each group). During inter-block intervals (12 seconds) the participants rated how causal each button was in earning rewards. These self-reports (causal ratings) are not used in the modelling analysis presented here.

Computational models

Notation. The set of available actions is denoted by $\mathcal{A}$ and the total number of available actions is denoted by N_a. Here $\mathcal{A} = \{\mathrm{L}, \mathrm{R}\}$, with L and R referring to left and right key presses respectively ($N_a = 2$). A set of subjects is denoted by $\mathcal{S}$, and the total number of trials completed by subject $s \in \mathcal{S}$ over the whole task (all blocks) is denoted by $\mathcal{T}_s$. a_t^s denotes the action taken by subject s at trial t. The reward earned at trial t is denoted by r_t, and we use a_t to refer to an action taken at time t, either by the subjects or the models (in simulations).

Recurrent neural network model (RNN). Architecture. The architecture used is based on a recurrent neural network model (RNN) and is depicted in Fig 1. The model is composed of an LSTM layer [Long short-term memory; 29] and an output softmax layer with two nodes (since there are two actions in the task). The inputs to the LSTM layer are the previous action (a_{t-1} coded using one-hot transformation) and the reward received after taking action ($r_{t-1} \in \{0, 1\}$). The outputs of the softmax are probabilities of selecting each action, which are denoted by $\pi_t(a; \mathrm{RNN})$ for action $a \in \mathcal{A}$ at trial t.

The LSTM layer is composed of a set of LSTM cells (N_c cells). Each cell is associated with (i) a cell state denoted by c_t^k for cell k at time t, and (ii) cell output denoted by h_t^k for cell k at time t. Cell states and outputs are initially zero and are updated after receiving each input. Let's define

$\mathbf{c}_t = [c_t^1, \ldots, c_t^{N_c}]^T$ as a vector containing cell states for all the cells at time t ($\mathbf{c}_t \in \mathbb{R}^{N_c}$), and $\mathbf{h}_t = [c_t^1, \ldots, c_t^{N_c}]^T$ as a vector containing all the cell outputs at time ($\mathbf{h}_t \in \mathbb{R}^{N_c}$). Furthermore, assume that $\mathbf{x}_t$ is a vector containing inputs to the network at time t, i.e., one-hot representation of a_t and r_t ($\mathbf{x}_t \in \mathbb{R}^{N_a+1}$). The update rules for $\mathbf{c}_t$ and $\mathbf{h}_t$ are as follows:

$$\mathbf{f}_t = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (\mathbf{f}_t \in \mathbb{R}^{N_c}) \tag{1}$$

$$\mathbf{i}_t = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad (\mathbf{i}_t \in \mathbb{R}^{N_c}) \tag{2}$$

$$\mathbf{o}_t = \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) \quad (\mathbf{o}_t \in \mathbb{R}^{N_c}) \tag{3}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c) \quad (\mathbf{c}_t \in \mathbb{R}^{N_c}) \tag{4}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (\mathbf{h}_t \in \mathbb{R}^{N_c}), \tag{5}$$

in which σ refers to the sigmoid function and $\odot$ represents the element-wise Hadamard product. The parameters of the LSTM layer include $W \in \mathbb{R}^{N_c \times (N_a+1)}$, $U \in \mathbb{R}^{N_c \times N_c}$, and $\mathbf{b} \in \mathbb{R}^{N_c}$.

The softmax layer takes outputs from the LSTM layer as its inputs ($\mathbf{h}_t$) and provides the probability of selecting each action π_t (a; RNN). The parameter of the softmax layer is $V \in \mathbb{R}^{N_c \times N_a}$, and therefore the parameters of the RNN model will be $\Theta = \{V, W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c, \mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_c\}$.

Training. In the training phase (learning how humans learn), the aim is to train weights in the network so that the model learns to predict subjects' actions given their past observations (i.e., it learns how *they* learn). This can be thought as a variant of a *learning-to-learn* process; albeit, more commonly, the learner is a human facing a series of learning tasks rather than a computer model trying to copy the human on a single task. For this purpose, the objective function for optimising weights in the network(denoted by Θ) for subject set $\mathcal{S}$ is,

$$\mathcal{L}(\Theta; \text{RNN}) = \sum_{s \in \mathcal{S}} \sum_{t=1 \cdots T_s} \log \pi_t(a_t^s; \text{RNN}), \tag{6}$$

where a_t^s is the action selected by subjects s at trial t, and π_t (.; RNN) is the probability that model assigns to each action. Note that the policy is conditioned on the previous actions and rewards in each block of training; notation for this is omitted, for simplicity.

Models were trained using the maximum-likelihood (ML) estimation method,

$$\Theta_{\text{RNN}}^{\text{ML}} = \arg \max_{\Theta} \mathcal{L}(\Theta; \text{RNN}), \tag{7}$$

where Θ is a vector containing free-parameters of the model (in both LSTM and softmax layers). The models were implemented in TensorFlow [61] and optimized using Adam optimizer [62]. Note that Θ was estimated for each group of subjects separately. Networks with different numbers N_c of LSTM cells ($N_c \in \{5, 10, 20\}$) were considered, and the best model was selected using leave-one-out cross-validation (see below). Early stopping was used for regularization and the optimal number of training iterations was selected using leave-one-out cross-validation.

The total number of free parameters (in both the LSTM layer and softmax layer) were 190, 580, and 1960 for the networks with 5, 10, and 20 LSTM cells, respectively. In order to control for the effect of initialization of network weights on the final results, a single random network of each size (5, 10, 20) was generated, and was used to initialize the weights in the network.

After the training phase, the weights in the network were frozen and the trained model was used for three purposes: (i) cross-validation (see below), (ii) on-policy simulations and (iii) off-policy simulations. For cross-validation, the previous actions of the test subject(s) and the rewards experienced by the subject(s) were fed into the model, but unlike the training phase, the weights were not changing and we only recorded the prediction of the model about the next action. Note that even though the weights in the network were fixed, the output of the network changed from trial to trial due to the recurrent nature of these networks.

Due to the small sample size, we used the same set of subjects for testing the model and for the validation of model hyper-parameters (N_c and number of optimization iterations). That is, we calculated the prediction accuracy of the model in each group using cross-validation for different numbers of training iterations and different numbers of cells (S1 Fig) and chose the hyper-parameters (N_c and number of optimization iterations) that led to the highest performance (for the comparison with other models). Another alternative to this in-sample hyper-parameter selection was to use the data from two of the groups to obtain the optimal hyper-parameters (number of iterations/cells) for the other group. We found that the prediction accuracies obtained using these two alternatives were similar across the groups. The results reported in the paper are those derived using the estimations based on in-sample hyper-parameter estimations. S17 Fig shows cross-validation results using the alternative hyper-parameter estimation (using other groups for the estimations) and S10 Table shows the comparison of the cross-validation results obtained using the two methods.

Other than being used for calculating cross-validation statistics, trained models were used for on-policy and off-policy simulations (with frozen weights). In the on-policy simulations, the model received its own actions and earned rewards as inputs (instead of receiving the action selected by the subjects). In the off-policy simulations, the set of actions and rewards that the model received was fixed and predetermined. The details of these simulations are reported in the Results section.

Model settings. For the RNN model, leave-one-out cross-validation was used to determine the number of cells and optimisation iterations required for the RNN model to achieve the highest prediction accuracy. We found that the lowest mean negative log-probability (NLP) was achieved by 10 cells in the LSTM layer and after 1100, 1200 optimisation iterations for the HEALTHY and DEPRESSION groups respectively whereas for the BIPOLAR group the best NLP was achieved by 20 cells and 400 optimisation iterations (see S1 Fig). These settings were used for making predictions and simulations.

Baseline methods. For our baselines, we used QL, QLP and GQL– which are variants and generalizations of *Q*-learning [31]—and LIN, which is a logistic regression model.

QL *model*. After taking action a_{t-1} at time $t-1$, the value of the action, denoted by $Q_t(a_{t-1})$, is updated as follows,

$$Q_t(a_{t-1}) = (1-\phi)Q_{t-1}(a_{t-1}) + \phi r_{t-1}, \tag{8}$$

where ϕ is the learning-rate and r_{t-1} is the reward received after taking the action. Given the action values, the probability of taking action $a \in \{\mathrm{L}, \mathrm{R}\}$ in trial t is:

$$\pi_t(a; \mathrm{QL}) = \frac{e^{\beta Q_t(a)}}{\sum_{a' \in \mathcal{A}} e^{\beta Q_t(a')}},$$

where $\beta > 0$ is a free-parameter and controls the contribution of values to the choices (balance between exploration and exploitation). The free-parameters of this variant are ϕ and β. Note that the probability that the models predict for each action at trial t is necessarily based on the data *before* observing the action and reward at trial t. Further, since there are only two actions,

we can write $\pi_t(\mathrm{L};\ \mathrm{QL}) = 1 - \pi_t(\mathrm{R};\ \mathrm{QL}) = \sigma(\beta(Q_t(\mathrm{L}) - Q_t(\mathrm{R})))$ where $\sigma(\cdot)$ is the standard logistic sigmoid.

Note that since here we are focused on modeling a bandit task, Q-values are represented as a function of actions and not states.

QLP *model*. This model is inspired by the fact that humans and other animals have a tendency to stick with the same action for multiple trials (i.e., perseverate), or sometimes to alternate between the actions [independent of the reward effects; 33]. We therefore call this model QLP, for Q-learning with perseveration. In it, action values are updated according to Eq 8 and so similarly to the QL model, but the probability of selecting actions is,

$$\pi_t(a;\mathrm{QLP}) = \frac{e^{\beta Q_t(a)+k_t(a)}}{\sum_{a'\in\mathcal{A}} e^{\beta Q_t(a')+k_t(a')}},$$

where,

$$k_t(a) = \begin{cases} \kappa & \text{if } a = a_{t-1} \\ 0 & \text{otherwise} \end{cases}. \tag{9}$$

Therefore, there is a tendency to select the same action again on the next trial (if $\kappa > 0$) or switch to the other action (if $\kappa < 0$). In the specific case that $\kappa = 0$, the QLP model reduces to QL. Free-parameters are ϕ, β, κ.

GQL *model*. As we will show in the results section, neither QL nor QLP fit the behaviour of the subjects in the task. As such, we aimed to develop a baseline model which could at least capture high-level behavioural trends, and we built a generalised Q-learning model, GQL, to compare with RNN. In this variant, instead of learning a single action value for each action, the model learns d different values for each action, where the difference between the values learned for each action is that they are updated using different learning-rates. The action values for action a are denoted by $\mathbf{Q}(a)$, which is a vector of size d, and the corresponding learning-rates are denoted by vector Φ of size d ($\mathbf{0} \preceq \Phi \preceq \mathbf{1}$). Based on this, the value of action a_{t-1} at trial $t-1$ is updated as follows,

$$\mathbf{Q}_t(a_{t-1}) = (\mathbf{1} - \Phi) \odot \mathbf{Q}_{t-1}(a_{t-1}) + r_{t-1}\Phi, \tag{10}$$

where as mentioned before $\odot$ represents the element-wise Hadamard product. For example, if $d = 2$, and $\Phi = [0.1, 0.05]$, then the model will learn two different values for each action (L, R actions) with one of the values updated using a learning-rate of 0.1 and the other updated using a learning-rate of 0.05. In the specific case that $d = 1$, the above equation reduces to Eq 8 used in QL and QLP models, in which only a single value is learned for each action.

In the QLP model, the current action is affected by the last taken action (perseveration). This property is generalised in the GQL model by learning the history of previously taken actions instead of just the last action. These action histories are denoted by $\mathbf{H}(a)$ for action a. $\mathbf{H}(a)$ is a vector of size d, and each entry of this vector tracks the tendency of taking action a in the past, i.e., if an element of $\mathbf{H}(a)$ is close to one it means that action a was taken frequently in the past and being close to zero implies that the action was taken rarely. In similar fashion to action values, for each action d different histories are tracked, each of which is modulated by a separate learning-rate. Learning-rates are represented in vector Ψ of size d ($\mathbf{0} \preceq \Psi \preceq \mathbf{1}$). Assuming that action a_{t-1} was taken at trial $t-1$, $\mathbf{H}(a)$ updates as follows,

$$\mathbf{H}_t(a) = \begin{cases} (\mathbf{1} - \Psi) \odot \mathbf{H}_{t-1}(a) + \Psi & \text{if } a = a_{t-1} \\ (\mathbf{1} - \Psi) \odot \mathbf{H}_{t-1}(a) & \text{otherwise} \end{cases}. \tag{11}$$

Intuitively, according to the above equation, if action a was taken on a trial, $\mathbf{H}(a)$ increases (the amount of increase depends on the learning-rate of each entry), and for the rest of the actions, $\mathbf{H}$(other actions) will decrease (again the amount of decrement is modulated by the learning rates). For example, if $d = 2$, and $\Psi = [0.1, 0.05]$, it means that for each action two choice tendencies will be learned, one of which is updated by rate 0.1 and the other one by rate 0.05.

Having learned $\mathbf{Q}(a)$ and $\mathbf{H}(a)$ for each action, the next question is how are they combined to guide choice. Q-learning models assume that the contribution of values to choices is modulated by parameter β. Here, since the model learns multiple values for each action, we assume that each value is weighted by a separate parameter, denoted by vector $\mathbf{B}$ of size d. Similarly, in the QLP model the contribution of perseveration to choices is controlled by parameter κ, and here we assume that parameter $\mathbf{K}$ modulates the contribution of previous actions to the current choice. Based on this, the probability of taking action a at trial t is,

$$\pi'_t(a; \mathrm{GQL}) = \frac{e^{\mathbf{B}\cdot\mathbf{Q}_t(a)+\mathbf{K}\cdot\mathbf{H}_t(a)}}{\sum_{a'\in\mathcal{A}} e^{\mathbf{B}\cdot\mathbf{Q}_t(a')+\mathbf{K}\cdot\mathbf{H}_t(a')}},$$

where "$\cdot$" operator refers to the inner product. Here, we also add extra flexibility to the model by allowing values to interact with the history of previous actions in influencing choices. For example, if $d = 2$, we allow the two learned values for each action to interact with the two learned action histories of each action, leading to four interaction terms, and the contribution of each interaction term to choices is determined by a matrix $\mathbf{C}$ of size $d \times d$ ($d = 2$ in this example),

$$\pi_t(a; \mathrm{GQL}) = \frac{e^{\mathbf{B}\cdot\mathbf{Q}_t(a)+\mathbf{K}\cdot\mathbf{H}_t(a)+\mathbf{H}_t(a)\cdot\mathbf{C}\cdot\mathbf{Q}_t(a)}}{\sum_{a'\in\mathcal{A}} e^{\mathbf{B}\cdot\mathbf{Q}_t(a')+\mathbf{K}\cdot\mathbf{H}_t(a')+\mathbf{H}_t(a')\cdot\mathbf{C}\cdot\mathbf{Q}_t(a')}}, \tag{12}$$

The free-parameters of this model are Φ, Ψ, $\mathbf{B}$, $\mathbf{K}$, and $\mathbf{C}$. In this paper we use models with $d = 1, 2, 10$, which have 5, 12 and 140 free parameters respectively. We used $d = 2$ for the results reported in the main text, since this model setting was able to capture several behavioural trends while still being interpretable. The results using $d = 1, 10$ are reported in the supplementary materials to illustrate the models' capabilities in extreme cases.

LIN *model*. The probability of taking each action is determined by a history past rewards and actions—up to J trials back—using a linear logistic regression model,

$$\log\frac{\pi_t(a = \mathrm{L}; \mathrm{LIN})}{\pi_t(a = \mathrm{R}; \mathrm{LIN})} = \begin{cases} \mu_0 + \sum_{j=1}^{J} \mu_j a_{t-j} + \gamma_j r_{t-j} + \zeta_j a_{t-j} r_{t-j} & J > 0 \\ \mu_0 & J = 0 \end{cases}. \tag{13}$$

Parameter J was selected using cross-validation (S2 Fig), which indicated that $J = 18$ provides the best NLP mean, and therefore the model with $J = 18$ was used in the analyses presented in the paper.

Objective function. The objective function for optimising the models was the same as the one chosen for RNN,

$$\mathcal{L}(\Theta; \mathcal{M}) = \sum_{s\in\mathcal{S}} \sum_{t=1\ldots T_s} \log \pi_t(a_t^s; \mathcal{M}), \mathcal{M} \in \{\mathrm{QL}, \mathrm{QLP}, \mathrm{GQL}, \mathrm{LIN}\}, \tag{14}$$

where, as mentioned before, a_t^s is the action selected by subject s at trial t, and $\pi_t(.; \mathcal{M})$ is the probability that model $\mathcal{M}$ assigns to each action. Models were trained using the maximum-

likelihood estimation method,

$$\Theta_{\mathcal{M}}^{\mathrm{ML}} = \arg\max_{\Theta} \mathcal{L}(\Theta; \mathcal{M}), \tag{15}$$

where Θ is a vector containing the free-parameters of the models. Optimizations for all models except LIN, were performed using Adam optimizer [62], and using the automatic differentiation method provided in TensorFlow [61]. The free-parameters with limited support (ϕ, β, Φ, Ψ) were transformed to satisfy the constraints. For LIN model, we used 'glm' method in R [63] with 'binomial' link function to estimate the parameters and to make predictions.

Performance measures. Two different measures were used for quantifying the predictive accuracy of the models. The first measure is the average log-probability of the models' prediction for the actions taken by subjects. For a group of subjects denoted by $\mathcal{S}$, we define negative log-probability (NLP) as follows:

$$\mathrm{NLP} = -\frac{\sum_{s\in\mathcal{S}}\sum_{t=1\ldots\mathcal{T}_s}\log \pi_t(a_t^s; \mathcal{M})}{\sum_{s\in\mathcal{S}}\mathcal{T}_s}, \mathcal{M} \in \{\mathrm{RNN}, \mathrm{LIN}, \mathrm{GQL}, \mathrm{QL}, \mathrm{QLP}\}. \tag{16}$$

The other measure is the percentage of actions predicted correctly,

$$\%\mathrm{correct} = \frac{\sum_{s\in\mathcal{S}}\sum_{t=1\ldots\mathcal{T}_s}[\![\arg\max_a \pi_t(a; \mathcal{M}) = a_t^s]\!]}{\sum_{s\in\mathcal{S}}\mathcal{T}_s}, \tag{17}$$

where $[\![.]\!]$ denotes the indicator function. Unlike, '%correct', NLP takes the probabilities of predictions into account instead of making binary predictions for the next action. In this way, if the models are certain about wrong predictions NLP performance gets penalized, and it gets credit if the models are certain about a correct prediction.

Model selection. Leave-one-out cross-validation was used for comparing different models. At each round, one of the subjects was withheld and the model was trained using the remaining subjects; the trained model was then used to make predictions about the withheld subject. The withheld subject was rotated in each group, yielding 34, 34 and 33 prediction accuracy measures in the HEALTHY, DEPRESSION, and BIPOLAR groups respectively.

Statistical analysis

For the analysis we performed hierarchical linear mixed-effects regression using the lme4 package in R [64] and obtained p-values for regression coefficients using the lmerTest package [65]. Hierarchical mixed-effects models involve random-effects and fixed-effects. Fixed-effects are of primary interest and are estimated directly (fixed-effects estimates are denoted by η). Random-effects specify different levels at which the data is collected (e.g., different subjects), i.e., fixed-effects are nested within random-effects in a hierarchical manner. Specific fixed-effects and random-effects used for each analysis are mentioned below for each analysis. For each test we report parameter estimate (η), standard error (SE), and p-value.

For the analysis presented in section 'Performance in the task' the intercept term was the random-effect at the subject level; action (low reward probability = 0, high reward probabilities = 1) was the fixed-effect; the dependent variable was the probability of selecting the action. For the second set of analyses in this section, the intercept term was the random-effect at the subject level; and action (low reward probability = 0, high reward probabilities = 1), groups (HEALTHY = 0, DEPRESSION = 1/BIPOLAR = 1) and their interaction were fixed-effects; the dependent variable was the probability of selecting the action.

In the analysis in section 'The immediate effect of reward on choice' the intercept was the random-effect at the subject level; whether reward was earned on the previous trial was the fixed-effect and the probability of staying on the same action was the dependent variable.

In the analysis presented in section 'Action prediction' the intercept term was the random-effect at the cross-validation fold level; model (GQL = 1, QLP = 0 for the first analysis and LIN = 1, RNN = 0 for the second analysis) was the fixed-effect. NLP was the dependent variable.

In the analysis presented in section 'The effect of reward on choice' the intercept was the random-effect at the subject level; whether zero rewards or more than two rewards were earned previously was fixed-effect. The dependent variable was the probability ofstaying with an action.

In the analysis presented in section 'Task', the intercept term was the random-effect at the group level (HEALTHY or BIPOLAR), and the mode of task completion (fMRI setting = 1, computer = 0) was the fixed-effect; the probability of selecting the better key was the dependent variable.

In the analysis presented in section 'The effect of reward on choice' the intercept was the random-effect at the subject level; the number of times that an action was repeated since switching to the action was the fixed-effect (between zero to 15 times). The dependent variable was the probability of staying with an action. Note that in Fig 8:right-panel in this section, to be consistent with off-policy simulations, only trials on which (i) subjects did not earn a reward on that trial, and (ii) subjects did not earn a reward since switching to the current action, were included in the graph.

For Loess regression [66], 'loess' method in R was used [63].

Supporting information

S1 Text. Behavioural analysis using GQL.
(PDF)

S2 Text. The choice of off-policy settings.
(PDF)

S3 Text. Analysis of randomness of choices.
(PDF)

S1 Fig. Cross-validation results for different numbers of cells and optimization iterations. (Top-panel) Percentage of actions predicted correctly averaged over leave-one-out cross-validation folds. **(Bottom-panel)** Mean NLP averaged over cross-validation folds. Error-bars represent 1SEM.
(PDF)

S2 Fig. Cross-validation results for the LIN model as a function of number of trials back (*J*). (Left-panel) NLP (negative log-probability) averaged across leave-one-out cross-validation folds. Lower values are better. **(Right-panel)** Percentage of actions predicted correctly averaged over cross-validation folds. Error-bars represent 1SEM.
(PDF)

S3 Fig. Choices of the HEALTHY group. Each row shows the choices of a subject across different blocks (12 blocks).
(PDF)

S4 Fig. Choices of the DEPRESSION group. Each row shows the choices of a subject across different blocks (12 blocks).
(PDF)

S5 Fig. Choices of the BIPOLAR group. Each row shows the choices of a subject across different blocks (12 blocks).
(PDF)

S6 Fig. Off-policy simulations of LIN. Each panel shows a simulation for 30 trials (horizontal axis), and the vertical axis shows the predictions for each group on each trial. The ribbon below each panel shows the action which was fed to the model on each trial. In the first 10 trials, the action that the model received was R and in the next 20 trials it was L. Rewards are shown by black crosses (x) on the graphs. See text for the interpretation of the graph. Note that the simulation conditions are same as those depicted in Figs 7 and 6.
(PDF)

S7 Fig. Off-policy simulations of GQL ($d = 2$). Each panel shows a simulation for 30 trials (horizontal axis), and the vertical axis shows the predictions for each group on each trial. The ribbon below each panel shows the action which was fed to the model on each trial. In the first 10 trials, the action that the model received was R and in the next 20 trials it was L. Rewards are shown by black crosses (x) on the graphs. See text for the interpretation of the graph. Note that the simulation conditions are the same as those depicted in Figs 7 and 6.
(PDF)

S8 Fig. Off-policy simulations of QLP. Each panel shows a simulation for 30 trials (horizontal axis), and the vertical axis shows the predictions for each group on each trial. The ribbon below each panel shows the action which was fed to the model on each trial. In the first 10 trials, the action that the model received was R and in the next 20 trials it was L. Rewards are shown by black crosses (x) on the graphs. See text for the interpretation of the graph. Note that the simulation conditions are the same as those depicted in Figs 7 and 6.
(PDF)

S9 Fig. Off-policy simulations of QL. Each panel shows a simulation for 30 trials (horizontal axis), and the vertical axis shows the predictions for each group on each trial. The ribbon below each panel shows the action which was fed to the model on each trial. In the first 10 trials, the action that the model received was R and in the next 20 trials it was L. Rewards are shown by black crosses (x) on the graphs. See text for the interpretation of the graph. Note that the simulation conditions are the same as those depicted in Figs 7 and 6.
(PDF)

S10 Fig. Off-policy simulations of GQL with $d = 1$. Each panel shows a simulation for 30 trials (horizontal axis), and the vertical axis shows the predictions for each group on each trial. The ribbon below each panel shows the action which was fed to the model on each trial. In the first 10 trials, the action that the model received was R and in the next 20 trials it was L. Rewards are shown by black crosses (x) on the graphs. See text for the interpretation of the graph. Note that the simulation conditions are the same as those depicted in Figs 7 and 6.
(PDF)

S11 Fig. The effect of the initialisation of the network on the off-policy simulations of RNN. The simulation conditions are the same as those depicted in Figs 7 and 6. Here, 15 different initial networks were generated and optimised and the policies of the models on each trial were averaged. The grey ribbon around the policy shows the standard deviation of the policies.

Each panel shows a simulation for 30 trials (horizontal axis), and the vertical axis shows the predictions of each model on each trial. The ribbon below each panel shows the action which was fed to the model on each trial. In the first 10 trials, the action that the model received was R and in the next 20 trials it was L. Rewards are shown by black crosses (x) on the graphs. See text for the interpretation of the graph.
(PDF)

S12 Fig. Percentage of each run of actions relative to the total number of runs for each subject. Percentage of each length of run of actions relative to the total number of runs in each subject (averaged over subjects). Red dots represent data for each subject, and error-bars represent 1SEM.
(PDF)

S13 Fig. RNN simulations. The graph is similar to Fig 8 but using data from RNN simulations (on-policy). **(Left-panel)** Probability of staying with an action after earning reward as a function of the number of actions taken since switching to the current action (averaged over subjects). Each red dot represents the data for each subject. **(Right-panel)** Probability of staying with an actions as a function of the number of actions taken since switching to the current action. The red line was obtained using Loess regression (Local Regression), which is a non-parametric regression approach. The grey area around the red line represents 95% confidence interval. Error-bars represent 1SEM.
(PDF)

S14 Fig. LIN simulations. The graph is similar to Fig 8 but using data from LIN simulations (on-policy). **(Left-panel)** Probability of staying with an action after earning reward as a function of the number of actions taken since switching to the current action (averaged over subjects). Each red dot represents the data for each subject. **(Right-panel)** Probability of staying with an actions as a function of the number of actions taken since switching to the current action. The red line was obtained using Loess regression (Local Regression), which is a non-parametric regression approach. The grey area around the red line represents 95% confidence interval. Error-bars represent 1SEM.
(PDF)

S15 Fig. GQL simulations (d = 2). The graph is similar to Fig 8 but using data from GQL simulations with d = 2 (on-policy). **(Left-panel)** Probability of staying with an action after earning reward as a function of the number of actions taken since switching to the current action (averaged over subjects). Each red dot represents the data for each subject. **(Right-panel)** Probability of staying with an actions as a function of the number of actions taken since switching to the current action. The red line was obtained using Loess regression (Local Regression), which is a non-parametric regression approach. The grey area around the red line represents 95% confidence interval. Error-bars represent 1SEM.
(PDF)

S16 Fig. GQL simulations (d = 10). The graph is similar to Fig 9 but using data from GQL simulations with d = 10 (on-policy). Median number of actions executed in a row before switching to another action (run of actions) in each subject as a function of the length of the previous run of actions (averaged over subjects). The dotted line shows the points at which the length of the previous and current runs are the same. Note that the median rather than the average was because we aimed to illustrate the most common 'length of current run', instead of average run length in each subject. Error-bars represent 1SEM.
(PDF)

S17 Fig. Cross-validation results. (Left-panel) NLP (negative log-probability) averaged across leave-one-out cross-validation folds. Lower values are better. **(Right-panel)** Percentage of actions predicted correctly averaged over cross-validation folds. Note that the difference between this figure and Fig 5 is that in Fig 5 hyper-parameters were obtained using in-sample estimations but here we used the data from two of the groups to obtain the optimal hyper-parameters (number of iterations/cells) for the other group. See text for more information.
(PDF)

S1 Table. Prediction of diagnostic labels using LIN. Number of subjects for each true- and predicted-label. The numbers inside parentheses are the percentage of subjects relative to the total number of subjects in each diagnostic group.
(PDF)

S2 Table. Prediction of diagnostic labels using GQL ($d = 2$). Number of subjects for each true- and predicted-label. The numbers inside parentheses are the percentage of subjects relative to the total number of subjects in each diagnostic group.
(PDF)

S3 Table. Estimated parameters for QL model.
(PDF)

S4 Table. Estimated parameters for QLP model.
(PDF)

S5 Table. Estimated parameters for GQL model with $d = 2$.
(PDF)

S6 Table. Negative log-likelihood for each model optimized over all the subjects in each group.
(PDF)

S7 Table. Negative log-likelihood for each model. For RNN a single model was fitted to the whole group using ML estimation. For baseline methods (GQL, QLP, and QL), a separate model was fitted to each subject, and the reported number is the sum of negative log-likelihoods over the whole group.
(PDF)

S8 Table. Mean and standard deviation of negative log-likelihood for RNN over 15 different initialisations of the model and optimised over all the subjects in each group.
(PDF)

S9 Table. Average number of trials in each pair of reward probabilities in each group.
(PDF)

S10 Table. Mean of NLP derived using in-sample hyper-parameter estimation (in-sample) and using the data of other groups (other-groups).
(PDF)

S1 Data. Supporting data.
(ZIP)

Author Contributions

Conceptualization: Amir Dezfouli, Peter Dayan, Bernard W. Balleine.

Data curation: Amir Dezfouli, Kristi Griffiths.

Formal analysis: Amir Dezfouli, Kristi Griffiths, Fabio Ramos, Peter Dayan.

Funding acquisition: Bernard W. Balleine.

Investigation: Amir Dezfouli, Fabio Ramos, Bernard W. Balleine.

Methodology: Amir Dezfouli, Fabio Ramos.

Project administration: Bernard W. Balleine.

Resources: Bernard W. Balleine.

Software: Amir Dezfouli, Fabio Ramos.

Supervision: Peter Dayan, Bernard W. Balleine.

Writing – original draft: Amir Dezfouli.

Writing – review & editing: Kristi Griffiths, Fabio Ramos, Peter Dayan, Bernard W. Balleine.

References

1. Busemeyer JR, Diederich A. Cognitive modeling. Sage; 2010.
2. Daw ND. Trial-by-trial data analysis using computational models. In: Delgado MR, Phelps EA, Robbins TW, editors. Decision Making, Affect, and Learning. Oxford University Press; 2011.
3. Gold JI, Shadlen MN. The neural basis of decision making. Annual review of neuroscience. 2007; 30. https://doi.org/10.1146/annurev.neuro.29.051605.113038 PMID: 17600525
4. Piray P, Zeighami Y, Bahrami F, Eissa AM, Hewedi DH, Moustafa AA. Impulse control disorders in Parkinson's disease are associated withdysfunction in stimulus valuation but not action valuation. The Journal of neuroscience. 2014; 34(23):7814–24. https://doi.org/10.1523/JNEUROSCI.4063-13.2014 PMID: 24899705
5. Busemeyer JR, Stout JC. A contribution of cognitive decision models to clinical assessment:decomposing performance on the Bechara gambling task. Psychological assessment. 2002; 14(3):253–62 PMID: 12214432
6. Dezfouli A, Keramati MM, Ekhtiari H, Safaei H, Lucas C. Understanding Addictive Behavior on the Iowa Gambling Task UsingReinforcement Learning Framework. In: 30th Annual Conference of the Cognitive Science Society; 2007. p.1094–1099.
7. Montague PR, Dolan RJ, Friston KJ, Dayan P. Computational psychiatry. Trends in cognitive sciences. 2012; 16(1):72–80. https://doi.org/10.1016/j.tics.2011.11.018 PMID: 22177032
8. O'Doherty JP, Hampton A, Kim H. Model-based fMRI and its application to reward learning and decisionmaking. Annals of the New York Academy of sciences. 2007; 1104(1):35–53. https://doi.org/10.1196/annals.1390.022 PMID: 17416921
9. Miller KJ, Botvinick MM, Brody CD. Dorsal hippocampus contributes to model-based planning. Nature neuroscience. 2017; 20(9):1269. https://doi.org/10.1038/nn.4613 PMID: 28758995
10. Acuña DE, Schrater P. Structure Learning in Human Sequential Decision-Making. PLOS Computational Biology. 2010; 6(12):1–12.
11. Dayan P, Niv Y. Reinforcement learning: the good, the bad and the ugly. Current opinion in neurobiology. 2008; 18(2):185–96. https://doi.org/10.1016/j.conb.2008.08.003 PMID: 18708140
12. Siegelmann HT, Sontag ED. On the computational power of neural nets. Journal of computer and system sciences. 1995; 50(1):132–150. https://doi.org/10.1006/jcss.1995.1013
13. Song HF, Yang GR, Wang XJ. Reward-based training of recurrent neural networks for cognitive andvalue-based tasks. eLife. 2017; 6:1–24. https://doi.org/10.7554/eLife.21492
14. Zhang Z, Cheng Z, Lin Z, Nie C, Yang T. A neural network model for the orbitofrontal cortex and task spaceacquisition during reinforcement learning. PLOS Computational Biology. 2018; 14(1):e1005925. https://doi.org/10.1371/journal.pcbi.1005925 PMID: 29300746
15. Miconi T. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. eLife. 2017; 6:1–24. https://doi.org/10.7554/eLife.20899

16. Carnevale F, DeLafuente V, Romo R, Barak O, Parga N. Dynamic Control of Response Criterion in Premotor Cortex duringPerceptual Detection under Temporal Uncertainty. Neuron. 2015; 86(4):1067–1077. https://doi.org/10.1016/j.neuron.2015.04.014 PMID: 25959731

17. Mante V, Sussillo D, Shenoy KV, Newsome WT. Context-dependent computation by recurrent dynamics in prefrontalcortex. Nature. 2013; 503(7474):78–84. https://doi.org/10.1038/nature12742 PMID: 24201281

18. Song HF, Yang GR, Wang XJ. Training Excitatory-Inhibitory Recurrent Neural Networks forCognitive Tasks: A Simple and Flexible Framework. PLoS Computational Biology. 2016; 12(2):1–30. https://doi.org/10.1371/journal.pcbi.1004792

19. Barak O, Sussillo D, Romo R, Tsodyks M, Abbott LF. From fixed points to chaos: Three models of delayed discrimination. Progress in Neurobiology. 2013; 103:214–222. https://doi.org/10.1016/j.pneurobio.2013.02.002 PMID: 23438479

20. Yang GR, Song HF, Newsome WT, Wang XJ. Clustering and compositionality of task representations in a neuralnetwork trained to perform many cognitive tasks. bioRxiv. 2017; p. 183632.

21. Sussillo D, Churchland MM, Kaufman MT, Shenoy KV. A neural network that finds a naturalistic solution for theproduction of muscle activity. Nature Neuroscience. 2015; 18(7):1025–1033. https://doi.org/10.1038/nn.4042 PMID: 26075643

22. Hennequin G, Vogels TP, Gerstner W. Optimal control of transient dynamics in balanced networks supportsgeneration of complex movements. Neuron. 2014; 82(6):1394–1406. https://doi.org/10.1016/j.neuron.2014.04.045 PMID: 24945778

23. Rajan K, Harvey CDD, Tank DWW. Recurrent Network Models of Sequence Generation and Memory. Neuron. 2016; 90(1):128–142. https://doi.org/10.1016/j.neuron.2016.02.009 PMID: 26971945

24. Laje R, Buonomano DV. Robust timing and motor patterns by taming chaos in recurrent neuralnetworks. Nature Neuroscience. 2013; 16(7):925–933. https://doi.org/10.1038/nn.3405 PMID: 23708144

25. Hochreiter S, Younger AS, Conwell PR. Learning to learn using gradient descent. In: International Conference on Artificial Neural Networks. Springer;2001. p. 87–94.

26. Wang JX, Kurth-Nelson Z, Tirumala D, Soyer H, Leibo JZ, Munos R, et al. Learning to reinforcement learn. arXiv preprint arXiv:161105763. 2016.

27. Duan Y, Schulman J, Chen X, Bartlett PL, Sutskever I, Abbeel P. RL^2: Fast Reinforcement Learning via Slow ReinforcementLearning. arXiv preprint arXiv:161102779. 2016;.

28. Weinstein A, Botvinick MM. Structure Learning in Motor Control: A Deep Reinforcement LearningModel. arXiv preprint arXiv:170606827. 2017.

29. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997; 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 PMID: 9377276

30. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.

31. Watkins CJCH. Learning from Delayed Rewards [Ph. D. thesis]. Cambridge University; 1989.

32. Ito M, Doya K. Validation of decision-making models and analysis of decision variables in the rat basal ganglia. Journal of Neuroscience. 2009; 29(31):9861–74. https://doi.org/10.1523/JNEUROSCI.6157-08.2009 PMID: 19657038

33. Lau B, Glimcher PW. Dynamic response-by-response models of matching behavior in rhesusmonkeys. Journal of the experimental analysis of behavior. 2005; 84(3):555–79 PMID: 16596980

34. Kim H, Sul JH, Huh N, Lee D, Jung MW. Role of striatum in updating values of chosen actions. Journal of Neuroscience. 2009; 29(47):14701–14712. https://doi.org/10.1523/JNEUROSCI.2728-09.2009 PMID: 19940165

35. Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521(7553):436–444. https://doi.org/10.1038/nature14539 PMID: 26017442

36. Lake BM, Salakhutdinov R, Tenenbaum JB. Human-level concept learning through probabilistic programinduction. Science. 2015; 350(6266):1332–1338. https://doi.org/10.1126/science.aab3050 PMID: 26659050

37. Dezfouli A, Morris RW, Ramos F, Dayan P, Balleine BW. Integrated accounts of behavioral and neuroimaging data using flexible recurrent neural network models. In: Advances in Neural Information Processing Systems (Neurips);2018.

38. Karpathy A, Johnson J, Fei-Fei L. Visualizing and understanding recurrent networks. arXiv preprint arXiv:150602078. 2015.

39. Barak O. Recurrent neural networks as versatile tools of neuroscience research. Current Opinion in Neurobiology. 2017; 46:1–6. https://doi.org/10.1016/j.conb.2017.06.003 PMID: 28668365

40. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, et al. Intriguing properties of neural networks. arXiv preprint arXiv:13126199. 2013;.

41. Fründ I, Wichmann FA, Macke JH. Quantifying the effect of intertrial dependence on perceptualdecisions. Journal of vision. 2014; 14(7):9. https://doi.org/10.1167/14.7.9 PMID: 24944238

42. Howarth CI, Bulmer MG. Non-random sequences in visual threshold experiments. Quarterly Journal of Experimental Psychology. 1956; 8(4):163–171. https://doi.org/10.1080/17470215608416816

43. Lages M, Jaworska K. How predictable are "spontaneous decisions" and "hiddenintentions"? Comparing classification results based on previous responses with multivariate pattern analysis of fMRI BOLD signals. Frontiers in psychology. 2012; 3:56 PMID: 22408630

44. Lages M, Treisman M. A criterion setting theory of discrimination learning that accountsfor anisotropies and context effects. Seeing and perceiving. 2010; 23(5):401–434. https://doi.org/10.1163/187847510X541117 PMID: 21466134

45. Senders VL, Sowards A. Analysis of response sequences in the setting of a psychophysical experiment. The American journal of psychology. 1952; 65(3):358–374. https://doi.org/10.2307/1418758 PMID: 12976561

46. Treisman M, Williams TC. A theory of criterion setting with an application to sequentialdependencies. Psychological Review. 1984; 91(1):68.

47. Verplanck WS, Blough DS. Randomized stimuli and the non-independence of successive responsesat the visual threshold. The Journal of general psychology. 1958; 59(2):263–272. https://doi.org/10.1080/00221309.1958.9710195 PMID: 13587937

48. Angela JY, Cohen JD. Sequential effects: superstition or rational behavior? In: Advances in Neural Information Processing Systems (Neurips);2009. p. 1873–1880.

49. Wilder M, Jones M, Mozer MC. Sequential effects reflect parallel learning of multipleenvironmental regularities. In: Advances in Neural Information Processing Systems (Neurips);2009. p. 2053–2061.

50. Rush AJ, Trivedi MH, Wisniewski SR, Nierenberg AA, Stewart JW, Warden D, et al. Acute and longer-term outcomes in depressed outpatients requiring one or several treatment steps: A STAR*D report. American Journal of Psychiatry. 2006; 163(11):1905–1917. https://doi.org/10.1176/ajp.2006.163.11.1905 PMID: 17074942

51. Stephan KE, Penny WD, Daunizeau J, Moran RJ, Friston KJ. Bayesian model selection for group studies. NeuroImage. 2009; 46(4):1004–17. https://doi.org/10.1016/j.neuroimage.2009.03.025 PMID: 19306932

52. Piray P, Dezfouli A, Heskes T, Frank MJ, Daw ND. Hierarchical Bayesian inference for concurrent model fitting andcomparison for group studies. bioRxiv. 2018; p. 393561.

53. Akaike H. In: Parzen E, Tanabe K, Kitagawa G, editors. Information Theory andan Extension of the Maximum Likelihood Principle. New York, NY: Springer New York; 1998. p. 199–213.

54. Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A. Bayesian measures of model complexity and fit (with discussion). Journal of the Royal Statistical Society, Series B. 2002; 64:583–639(57). https://doi.org/10.1111/1467-9868.00353

55. van der Linde A. DIC in variable selection. Statistica Neerlandica. 2005; 59(1):45–56. https://doi.org/10.1111/j.1467-9574.2005.00278.x

56. Watanabe S. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. Journal of Machine Learning Research. 2010; 11(Dec):3571–3594.

57. Gelman A, Hwang J, Vehtari A. Understanding predictive information criteria for Bayesian models. Statistics and computing. 2014; 24(6):997–1016. https://doi.org/10.1007/s11222-013-9416-2

58. Hamilton M. A rating scale for depression. Journal of neurology, neurosurgery, and psychiatry. 1960; 23(1):56. https://doi.org/10.1136/jnnp.23.1.56 PMID: 14399272

59. Young RC, Biggs JT, Ziegler VE, Meyer DA. A rating scale for mania: reliability, validity and sensitivity. The British Journal of Psychiatry. 1978; 133(5):429–435. https://doi.org/10.1192/bjp.133.5.429 PMID: 728692

60. Goldman HH, Skodol AE, Lave TR. Revising axis V for DSM-IV: a review of measures of socialfunctioning. American Journal of Psychiatry. 1992; 149(9):1148–1156. https://doi.org/10.1176/ajp.149.9.1148 PMID: 1386964

61. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. Tensorflow: Large-scale machine learning on heterogeneousdistributed systems. arXiv preprint arXiv:160304467. 2016.

62. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:14126980. 2014.

63. R Core Team. R: A Language and Environment for Statistical Computing; 2016. Available from: https://www.r-project.org/.

64. Bates D, Mächler M, Bolker B, Walker S. Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software. 2015; 67(1):1–48. https://doi.org/10.18637/jss.v067.i01

65. Kuznetsova A, Bruun Brockhoff P, Haubo Bojesen Christensen R. lmerTest:Tests in Linear Mixed Effects Models; 2016.

66. Cleveland WS, Devlin SJ. Locally weighted regression: an approach to regression analysis bylocal fitting. Journal of the American statistical association. 1988; 83(403):596–610. https://doi.org/10.1080/01621459.1988.10478639