OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

Spike-based Decision Learning of Nash Equilibria in Two-Player Games

Johannes Friedrich, Walter Senn*

Department of Physiology and Center for Cognition, Learning and Memory, University of Bern, Switzerland

Abstract

Humans and animals face decision tasks in an uncertain multi-agent environment where an agent's strategy may change in time due to the co-adaptation of others strategies. The neuronal substrate and the computational algorithms underlying such adaptive decision making, however, is largely unknown. We propose a population coding model of spiking neurons with a policy gradient procedure that successfully acquires optimal strategies for classical game-theoretical tasks. The suggested population reinforcement learning reproduces data from human behavioral experiments for the blackjack and the inspector game. It performs optimally according to a pure (deterministic) and mixed (stochastic) Nash equilibrium, respectively. In contrast, temporal-difference(TD)-learning, covariance-learning, and basic reinforcement learning fail to perform optimally for the stochastic strategy. Spike-based population reinforcement learning, shown to follow the stochastic reward gradient, is therefore a viable candidate to explain automated decision learning of a Nash equilibrium in two-player games.

Citation: Friedrich J, Senn W (2012) Spike-based Decision Learning of Nash Equilibria in Two-Player Games. PLoS Comput Biol 8(9): e1002691. doi:10.1371/journal.pcbi.1002691

Editor: Olaf Sporns, Indiana University, United States of America

Received December 5, 2011; **Accepted** July 26, 2012; **Published** September 27, 2012

Funding: This work was supported by a grant of the Swiss SystemsX.ch initiative (Neurochoice, evaluated by the SNSF). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: senn@pyl.unibe.ch

Introduction

Neuroeconomics is an interdisciplinary research field that tries to explain human decision making in neuronal terms. Behavioral outcomes are construed as results of brain activity and the neuronal correlates of the quantities relevant for the decision making process are identified. Humans, as economic agents, attempt to optimize some reward function by participating in the production, exchange and maintenance of goods. Reward for the individuals will depend in general not merely upon their own actions but also on those of the other players and, furthermore, these will adapt their own strategies.

Classical models in neuroeconomics are based on temporal difference (TD) learning [1], an algorithm to maximize the total expected reward [2] with potential neuronal implementations [3,4]. It assumes that the environment can be described as a Markov decision process (MDP), i.e. by a finite number of states with fixed transition probabilities [5]. Multi-agent games, however, are not Markovian as the evolution of the environment typically does not only depend on the current state, but also on the history and on the adaptation of the other agents. Such games can be described as partially observable Markov decision processes (POMDP, [6]) by embedding the sequences and the learning strategies of the other agents into a large state space. We have presented a policy gradient method for population reinforcement learning which, unlike TD-learning, can cope with POMDPs and can be implemented in neuronal terms [7]. Yet, since a human learner would need to successfully explore the large state space of the POMDP, this appears to be an unrealistic scenario for explaining decision making in a multi-agent environment. A more realistic learning scenario is that humans transiently conceive the other players to follow a fixed strategy, and try to find their optimal counter strategy under this stationarity approximation. Maximizing one's own payoff while assuming stationarity in the opponents strategy is called a fictitious play and conditions are studied when this play effectively converges to a stationary (Nash) equilibrium [8].

Here we show that for classical two-player games [9] a simplified population reinforcement learning approach [7], which is policy gradient under the stationarity approximation, can reproduce human data. We consider two games, blackjack [10] and the inspector game [11], as examples for which the optimal strategy is either deterministic or stochastic, respectively. Optimality is expressed in terms of the Nash equilibrium, a solution concept for games involving two or more players. It is reached when no player has anything to gain by changing its strategy unilaterally. Each player is making the best decision it can, taking into account the decisions of the other(s), hence the Nash equilibrium constitutes an optimum. Our algorithm is consistent with behavioral experiments for these games [10,11] while performing optimally according to the Nash equilibrium. We also show that TD-learning as well as covariance learning fail to find the stochastic Nash equilibrium for the inspector game.

The current paper follows a long tradition of explaining human and animal behavior by simple models of reward-based learning, starting from Thorndike's law of effect [12] and Pavlovian conditioning paradigms [13,14] up to more recent theories of reinforcement learning [1,2,15,16]. Basic reinforcement learning with simple models of a few free parameters have also been applied to games. It has been shown for a wide set of two-player games that these simple algorithms well approximate human

Author Summary

Socio-economic interactions are captured in a game theoretic framework by multiple agents acting on a pool of goods to maximize their own reward. Neuroeconomics tries to explain the agent's behavior in neuronal terms. Classical models in neuroeconomics use temporal-difference(TD)-learning. This algorithm incrementally updates values of state-action pairs, and actions are selected according to a value-based policy. In contrast, policy gradient methods do not introduce values as intermediate steps, but directly derive an action selection policy which maximizes the total expected reward. We consider a decision making network consisting of a population of neurons which, upon presentation of a spatio-temporal spike pattern, encodes binary actions by the population output spike trains and a subsequent majority vote. The action selection policy is parametrized by the strengths of synapses projecting to the population neurons. A gradient learning rule is derived which modifies these synaptic strengths and which depends on four factors, the pre- and postsynaptic activities, the action and the reward. We show that for classical game-theoretical tasks our decision making network endowed with the four-factor learning rule leads to Nash-optimal action selections. It also mimics human decision learning for these same tasks.

performance [17].Yet, we show that basic reinforcement learning does not follow the reward gradient, and in fact it does not fit human data on the inspector game as well as our gradient rule. Obviously, playing games involves cognitive reasoning, as for instance captured by the theory of 'adaptive control of thought–rational' (ACT-R, [18]). Within such a theory, our model represents a neuronal implementation of a 'production rule' which initiates a behavioral pattern in response to sensory and possibly cognitive input.

Results

Model

The network representing a player consists of a population of $N=100$ Spike-Response-Model (SRM) neurons with escape noise [19], driven by a common presynaptic stimulus encoding the current state of the game (the player's hand value in blackjack, and a fixed stimulus for the inspector game). Each input spike pattern (X) is composed of $M=80$ afferent spike trains generated once by independent $6\,\mathrm{Hz}$ Poisson processes with $T=500\,\mathrm{ms}$ duration, and then repeatedly presented with the same fixed spike timings. The population neurons integrate the afferent presynaptic input spike trains and produce an output spike pattern (Y, see Fig. 1). The decision of an individual postsynaptic neuron is denoted by c, with $c=-1$, if the considered neuron does not spike, otherwise $c=1$. Behavioral decisions $D=\pm 1$ are stochastically made based on the population activity A defined as sum of the individual decisions c across the N population neurons: if A is small, the population decision is likely $D=-1$, and the larger A is, the more likely is $D=1$. At the end of a game involving either a single decision (like in the inspector game) or a sequence of decisions (like in blackjack), a reward signal R is delivered by an external critic which either informs about winning or losing (with $R=\pm 1$ like in blackjack) or delivers a specific payoff (like in the inspector game).

The synapses feeding the stimulating spike pattern to the population neurons are updated according to a multi-factor plasticity rule involving the reward, the behavioral decision, the single neuron decision and the eligibility trace which depends on the post- and pre-synaptic activity:

$$\Delta w = \mathrm{Rew}\,\mathrm{Dec}\,c\,E\ . \tag{1}$$

Here, Rew is the reward signal encoding the reward prediction error [20] (see Eq. 5 in Methods), Dec is the global feedback signal informing the synapses about the population decision D weighted by the population activity (Eq. 6 in Methods), and $c=\pm 1$ is the neuronal decision (spike/no spike). The eligibility trace E is a synaptic buffer roughly encoding the covariance between the past pre- and postsynaptic activity relevant for learning (Eq. 8 in Methods). Technically, E is the derivative of the log-likelihood of producing the postsynaptic spike train. The learning rule can be shown to perform gradient ascent in the expected reward (Supporting Text S2).

While most of the terms in Eq. 1 may have their standard biological counterpart [16], there is less experimental evidence for assigning the decision feedback Dec' to one specific neuromodulator. Yet, be the population neurons recurrently connected [21] or not, decision learning based on a population always requires that a global population signal is fed back to the individual neurons, as otherwise learning would quickly degrade with increasing population size [7,22]. By the same performance reasons it is not possible to replace the other factors '$\mathrm{Rew}\,c\,E$' in Eq. 1 by a classical spike-timing dependent plasticity (STDP) implementation endowed with the multiplicative reward signal 'Rew' [23]. In fact, reward-modulated STDP is only able to learn multiple stimulus-response associations when the reward factor averages out to zero for each stimulus individually, requiring an additional reward-prediction network [16].

Our neuronal implementation is as simple as possible to provide the required computational properties. The lack of feedback connectivity avoids issues relating to population spike correlations [24], and the neural mechanisms supporting the readout of the decision and the population feedback signal are not considered here. Similarly, the fixed spike trains representing an input pattern is a biological simplification which does not fundamentally restrict the suggested approach.

Blackjack

The simplified version of blackjack considered here was played in 18th century France and is the precursor of the version played in casinos nowadays. The card decks used consist of 52 cards. Ace counts eleven, jack, queen and king ten points and the numbers two to ten according to their written value. The player (gambler) draws one card after the other, starting with an initial two card hand, with the object of bringing the hand value (total across drawn cards) as close as possible to 21, but stopping early enough so that it does not exceed this number, in which case he immediately loses. Afterwards the croupier does the same for the bank. The player wins if its score is higher than that of the croupier or if the croupier exceeds 21, otherwise the croupier wins. The winner's payoff is 1, the loser's -1. We assume that both player and croupier base their decision whether to draw another card or not only on their current hand value. Player and bank follow a strategy defined by the hand value s_1 and s_2, respectively, from which on they stop to draw another card.

The described rules of the game result in the payoff-matrix (Table 1) comprising the average payoff of the bank as a function of the strategies s_1 and s_2 of the player and bank, respectively (Methods). The gambler loses whatever the bank wins, therefore the game is an example of a zero sum game. For zero sum games a Nash equilibrium corresponds to a minimax solution [25]. If the pay-off matrix has a saddle point (an entry which is the maximum

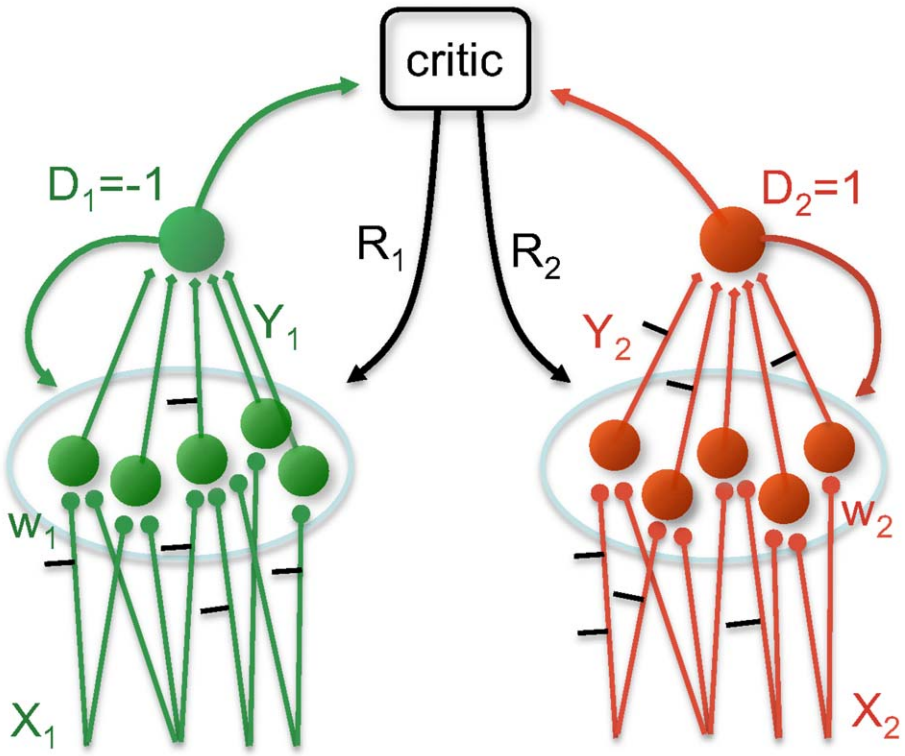

Figure 1. Neuronal architecture implementing the two players. Each player i (=1,2) is represented by a population of decision making neurons (shown 5 of each) which receive an input spike pattern X_i and generate an output spike pattern Y_i. The population decision $D_i = \pm 1$ is represented by a readout unit, with $D_i = 1$ being more likely when more decision making neurons fire at least one output spike. The synaptic weights w_i are adapted as a function of X_i, Y_i, D_i and the reward signal R_i delivered by a critic.
doi:10.1371/journal.pcbi.1002691.g001

in its row and the minimum in its column) the corresponding strategy pair is a minimax solution which represents a pure Nash equilibrium. In blackjack there is a unique such pair, $(s_1 = 15, s_2 = 16)$, and hence there is a unique Nash equilibrium at all. For this optimal strategy pair the gambler stops drawing another card as soon as he has 15 points or more, while the croupier stops at 16 or more. The entry represents the lowest loss for the gambler given the strategy of the bank (minimum in the column), and the maximal payoff obtainable by the bank given the strategy of the gambler (maximum in the row). The Nash equilibrium is asymmetric because in the case of a standoff (equal final hand values) the croupier always obtains reward 1 and the player -1. For hand values smaller than 11 it is safe to draw another card whereas for more than 19 drawing another card leads to certain loss due to exceeding 21. While we do not model these trivial actions, we address the learning problem for hand values between 11 and 19.

pRL and TD-learning converge to a pure Nash equilibrium. We first simulated two neural networks playing against each other. Each hand value between 11 and 19 was represented by a fixed spatio-temporal spike pattern generated by 6 Hz Poisson processes, with different (but fixed) patterns distinguishing numbers, gambler and croupier. Since initially no ordering information is associated to the Poisson spike train encoding of the hand values, the learning process has yet to assign this information by trial and error. The drawing probabilities for each hand value learned by the gambler after 500 and 10 000 games, averaged over ten runs, are shown in Fig. 2A. The colored dashed lines in the plot indicate the decision boundaries of Nash equilibrium above which no further card is drawn by the gambler or croupier, respectively. Initially, both players randomly decide with 50% chance to draw (black dashed line). After about 500 games both players have learned to not exceed 21 and do not draw further cards for high hand values, being still undetermined about what action to take for low hand values. After 10 000 games both have learned successfully to draw another card for low hand values and tend to play according to the Nash equilibrium.

We next simulated the gambler by a neural net and the croupier by a computer algorithm which follows right from the beginning a fixed strategy. The resulting drawing probabilities after 10 000 games are shown in Fig. 2B for three different strategies of the croupier, $s_2 = 15$, 16 and 17. The gambler learns the perfect response strategies. The neural net exploits deviations of the croupier from the Nash equilibrium $s_2 = 16$ by also deviating and thus increasing its reward. If the croupier stops earlier at $s_2 = 15$ the gambler continues until $s_1 = 16$, trying to exceed the croupier's hand value (blue), whereas if the croupier stops later at $s_2 = 17$ the gambler stops already at $s_1 = 12$ (green), taking advantage of the fact that the croupier likely exceeds 21. For the case $s_2 = 16$ (red), the gambler does not learn the optimal strategy as well as for the two others. This is due to the fact that here the true mean rewards for the strategies $s_1 = 14,16$ are close to the one for the optimal $s_1 = 15$, cf. Table 1, and cannot be distinguished based on merely 10 000 samples. Instead of just looking at the strategy we hence consider the maximally possible and the actually obtained reward for the three croupier strategies. Fig. 2C depicts the low pass filtered reward which approaches the theoretical optimum indicated by the dashed lines and read out from the corresponding columns $s_2 = 15,16,17$ in Table 1. For the non-optimal croupier strategies $(s_2 = 15,17)$ the maximally possible reward is significantly higher.

Table 1. Average bank payoff for our version of blackjack.

$s_1 \backslash s_2$	13	14	15	*16*	17	18	19
11	0.2982	0.3164	0.3027	*0.2544*	0.1689	0.0436	−0.1237
12	0.1635	0.2015	0.2076	*0.1791*	0.1130	0.0066	−0.1427
13	0.1052	0.1587	0.1806	*0.1679*	0.1176	0.0266	−0.1077
14	0.0438	0.1134	0.1536	*0.1597*	0.1282	0.0560	−0.0598
15	*0.0119*	*0.0706*	*0.1289*	**0.1555**	*0.1450*	*0.0940*	*−0.0008*
16	0.0143	0.0607	0.1085	*0.1557*	0.1685	0.1411	0.0702
17	0.0543	0.0893	0.1254	*0.1628*	0.1989	0.1980	0.1539
18	0.1349	0.1598	0.1854	*0.2120*	0.2394	0.2651	0.2509

The values show the calculated mean gains of the bank, respectively losses of the gambler, dependent on the strategy of the gambler (stopping to draw a card at hand values equal or larger than s_1) and the croupier (stopping at s_2). The strategies are described by the hand values from which on the players stop to draw another card. Formatted typesetting is used to highlight the Nash equilibrium.

doi:10.1371/journal.pcbi.1002691.t001

Hence, from playing against another network and against the croupier, we conclude that a neuronal population endowed with the plasticity rule (1) is able to learn the optimal strategies in blackjack, determined either by the pure Nash equilibrium, or by the croupier's fixed strategy. Replacement of the neural net by a TD-learner yields similar results for both scenarios (Supporting Text S1).

pRL fits human data on blackjack. Human behavior in blackjack was studied in [10,26], though with a deck of 32 cards. The instruction of the subjects about the rules of the game already induces a prior in the drawing behavior with a preference for drawing or stopping in the case of low or high hand values, respectively. Neither pRL nor TD can reproduce this type of learning by insight. Moreover, the network needs first to learn the ordering of the stimuli by trial-and-error, and hence much more learning trials are required for the network. To still allow for a comparison with the human data, we used the same deck of 32 cards and simulated a neural network with initial weights chosen in such a way, that the initial strategy mimics the one of human's (Fig. 2D). After playing the same number of games as humans did, the network's final strategy agrees with the experimental data of humans, showing the same shift to a slightly less risky drawing behavior.

Inspector game

The inspector game [27] has been widely studied in neuroeconomics [28]. The economic story surrounding the game is that a lazy employee prefers not to work. An employer knows this and sometimes 'inspects', but has to pay some cost '*i*'for inspection. The payoffs for employee and employer are shown in Table 2. The inspector game shows only a mixed Nash equilibrium in which decisions are taken stochastically with a fixed probability. At the equilibrium, the players mix pure strategies, each with the same payoff: had these pure strategies different payoffs, then it would be better to just follow the pure strategy with the highest expected payoff.

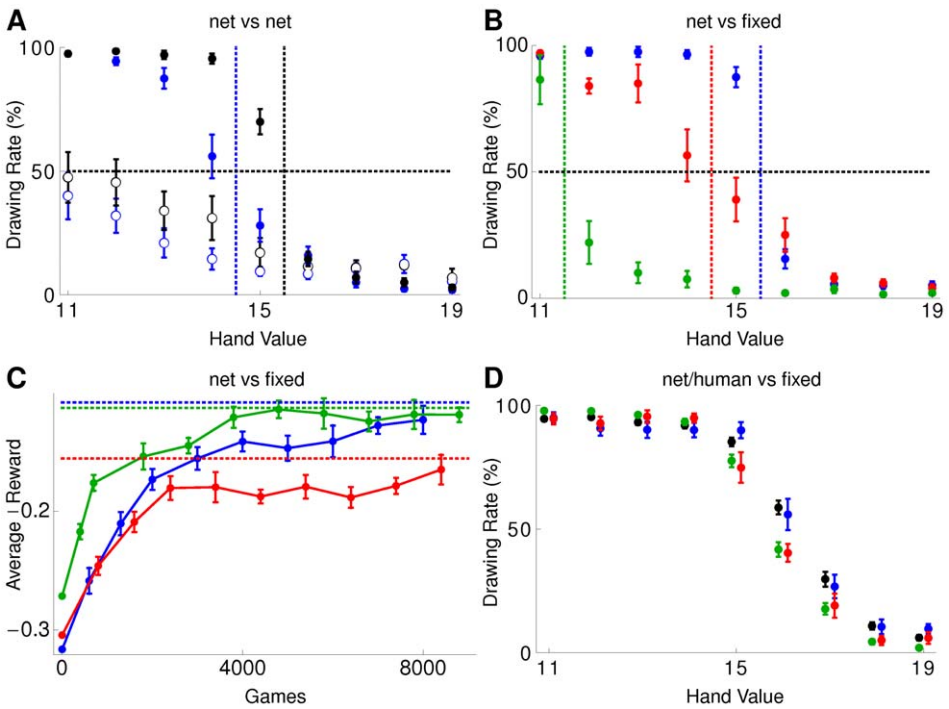

Figure 2. Playing blackjack with pRL converges toward pure Nash equilibrium. (A) Average strategy ($\pm$SEM) after 500 (open circles) and 10000 (filled circles) games where the gambler (blue) is a neural net as well as the croupier (black). The dotted vertical lines left of $s_1 = 15$ and $s_2 = 16$ show the separation line of drawing/not drawing another card for the optimal Nash strategy pair. (B) Average strategy ($\pm$SEM) after 10000 games for a neural net as gambler playing against a croupier that follows a given strategy $s_2 = 15$ (blue), 16 (red) or 17 (green). The colored dotted lines left of $s_1 = 12,15,16$ show the separation line of drawing/not drawing another card for the optimal strategy given that the croupier stops drawing at $s_2 = 17,16,15$ (from left to right). (C) Average reward ($\pm$SEM) of the gambler for the scenario described in (B). The colored dotted lines show the maximal reachable average reward. (D) Average strategy ($\pm$SEM) over the last 100 out of a total of 880 games for a neural net (red) or human (green) as gambler playing against a croupier that follows a given strategy $s_2 = 15$. The initial weights of the network were chosen such that the strategy in the first 100 trials (blue) mimics the strategy of humans instructed about the game rules (black).

doi:10.1371/journal.pcbi.1002691.g002

Table 2. Payoff matrix of the inspector game.

employee *employer*	*inspect*	*Don't inspect*
work	0.5, *2-i*	0.5, *2*
shirk	0, *1-i*	1, *0*

The variables in the left of each cell determine the employee's payoffs, and the variables in the right determine the employer's payoffs for each combination of player's responses. '*i*'is the cost of inspection to the employer.
doi:10.1371/journal.pcbi.1002691.t002

For each value of the inspection cost i, there is a unique mixed Nash equilibrium in which the probability with which the employee is shirking just corresponds to the inspection cost, $p(\text{shirk})=i$, and the probability with which the employer is inspecting is $p(\text{inspect})=0.5$. In this case, neither player can improve its expected payoff by only unilaterally changing its strategy. In fact, using $p(\text{shirk})=i$, the expected payoff for the employer is always

$$\begin{aligned}\langle R_2\rangle &= p(\text{shirk})p(\text{inspect})(1-i)+p(\text{work})p(\text{inspect})\\ &\quad (2-i)+p(\text{work})p(\text{don't inspect})2\\ &= ip(\text{inspect})(1-i)+(1-i)p(\text{inspect})(2-i)+\\ &\quad (1-i)(1-p(\text{inspect}))2=2(1-i),\end{aligned}$$

independently of $p(\text{inspect})$. Likewise, if $p(\text{inspect})=0.5$, the expected payoff for the employee is always $\langle R_1\rangle=0.5$, independently of $p(\text{shirk})$.

pRL reproduces inspector game data and Nash equilibria. We played with our neural network as employee against the algorithm presented in [11] as employer, and we also simulated two neural nets playing against each other. Fig. 3A shows a running average over the last 20 trials of the shirk and inspection rates for the neuronal net employee (green) and neuronal net employer (red), overlaid with the corresponding data for a human employee (black) playing against a human employer (grey, data from [11]). The inspection cost was held constant during three blocks of 150 trials and stepped from $i=0.5$ to 0.9 and finally to 0.3. The averaged shirk rate of the neuronal net employee is in striking agreement with the one of the human employee across the whole rate of inspection costs (Fig. 3B). There is also good agreement with the experimental data for the employee's reward (Fig. 3C).

To check whether the good fit of the human data is due to the gradient property alone or whether the population boost is also necessary we considered single neurons playing against each other (by setting the number of neurons in the population to $N=1$, without changing the learning rule). In this case our learning rule becomes equivalent to the policy gradient rule for single escape rate neurons [29,30]. With only a single neuron learning turns out to be too slow to match the transient behavior in the human data (Fig. 3A), even after optimizing the learning rate (data not shown). We have previously shown that the speeding up learning in a population of spiking neurons is only possible with an additional population signal modulating synaptic plasticity [22,31]. We conclude that population learning is necessary, and that other spike-based gradient rules which do not exploit a population signal [15,32,33] will also be too slow.

During the 150 trials across an experimental block, the shirk rates (but not the inspection rates) tended towards the corresponding value of the Nash equilibrium (diagonal line in Fig. 3B), and so did the employee's reward (horizontal line in Fig. 3C), although without reaching them. In the simulation we extended the block size to check the asymptotic behavior. We found that for block sizes of 5000 trials, the average shirk *and* the inspection rates closely reached the Nash equilibrium (match of simulation points with the two lines in Fig. 3D).

Despite the match of the average rates with the mixed Nash equilibria, the running means oscillate around the corresponding equilibria (shown in Fig. 3E for inspection cost $i=0.7$), as predicted by the theory [34,35]. In the asymmetric case, when our neuronal employee plays against the (apparently not optimal) computer algorithm, the oscillation vanish and the employee's shirk rate reaches the optimal Nash equilibrium (blue), as expected for a neuronal network endowed by synaptic modifications following the reward gradient (Supporting Text S2). When two reward maximizing networks play against each other, however, each tries to exploit any deviation of the other from his Nash equilibrium, pushing him even further away. This leads to oscillations around the Nash equilibrium where a change in strategy of one player is oppositely directed to the deviation from the Nash equilibrium of the other player. In fact, when superimposing one rate with the negative change of the other rate, a close match is observed (Fig. 3F).

As we are studying a policy gradient algorithm, one may ask how robust the described properties are in view of a possibly improper biological implementation. The eligibility trace E (Eq. 8 in Methods) depends on the presynaptic spike timing and contains a positive term that depends itself on the postsynaptic spike timing and a negative term depending on the postsynaptic potential. Due to the policy gradient property, the two terms are balanced and the eligibility trace is zero on average. To check for robustness we performed simulations where this balance is perturbed. The above results still qualitatively hold true if the negative term in the eligibility trace is twice as large, or even if it is neglected completely, yielding STDP with plasticity for pre-post spike pairing only. The robustness can be attributed to the factor Rew in the learning rule (Eq. 1) which averages out to 0 due to the subtraction of the reward prediction (since $\text{Rew}=R-\bar{R}$, see Eq. 5 in Methods) and hence neutralizes any bias in the estimate of E [16].

TD-learning is inconsistent with data and Nash. To value the match generated by our synaptic plasticity rule with experimental and theoretical data, we also trained a TD-learner on the inspector game. Yet, the parameter optimized TD-algorithm roughly reproduces only the humans average shirk rate (Fig. 3B), but less well the subjects' rewards (Fig. 3C). More strikingly, two opposing TD-learners simulated across the 5000 trial blocks do not behave according to the Nash equilibrium (Fig. 3D), but adopt a deterministic strategy within such a block (Fig. 3E). When simulating even longer, oscillations emerge as well, but without convergence of the long-term average to the Nash equilibrium.

There are principled reasons why TD-learning must generally fail to find an optimum solution, and in particular a mixed Nash equilibrium. First, TD-learning assumes that the underlying process is Markovian, but for multiplayer games this assumption is in general not satisfied. In fact, because the policy of the other player may change in time during learning, the optimal decision probabilities depend on past actions. Values which are assumed to be a function of the current action only, may therefore be incorrectly estimated. Second, for a mixed Nash equilibrium, TD-learning may also fail to correctly map values to decision probabilities at the steady-state after learning. This is because each policy imposes some predefined mapping from action values to decision probabilities, and this adhoc mapping may not reflect the true relationship between expected payoffs and decision probabilities defining the Nash equilibrium. For the inspector game with inspection

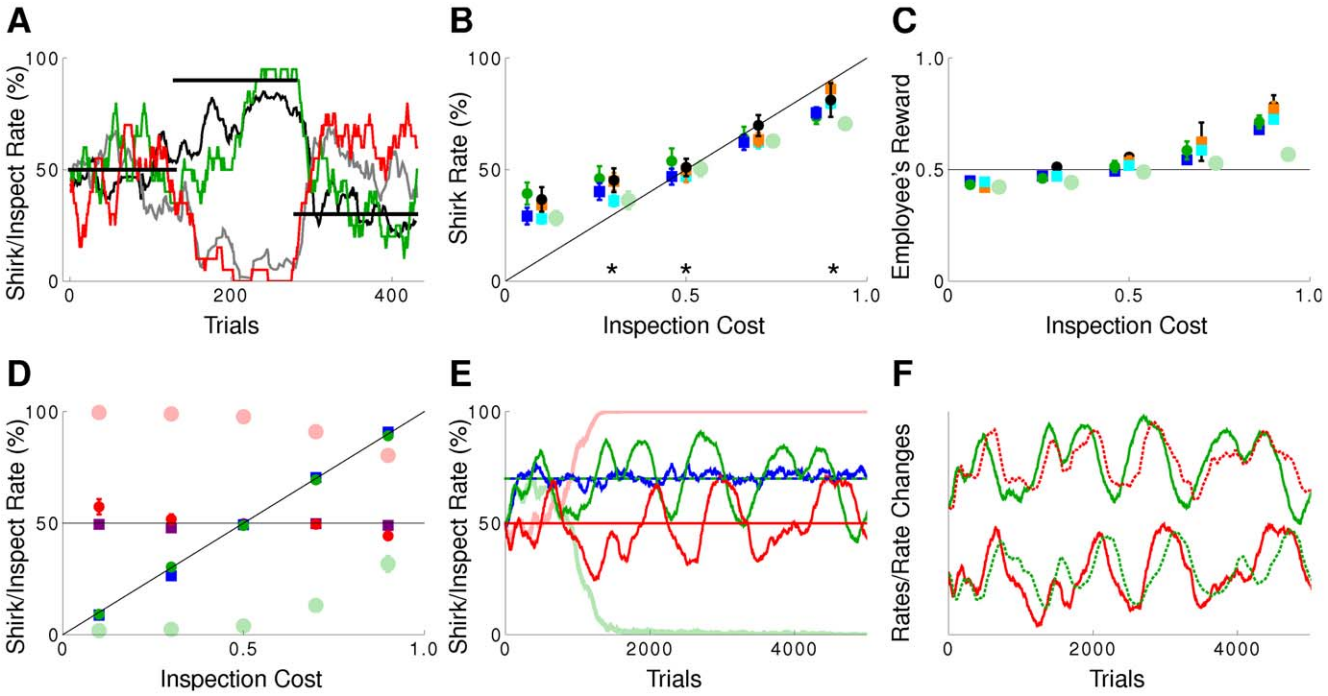

Figure 3. pRL but not TD-learning fits data and follows a mixed Nash equilibrium. (A) Choice behavior for pRL versus pRL (employee green, employer red) and human versus human (employee black, employer gray) [11]. The cost of inspection was stepped from 0.5 to 0.9 to 0.3, respectively, and this does also correspond to the shirk rate in Nash equilibrium (thick black lines). The inspection rate in the Nash equilibrium would always be 0.5. (B) Average choice behavior of pRL vs pRL (dark green circles) and TD vs TD (light green circles), pRL for the employee vs computer algorithm for the employer (blue squares), human vs human (black), human as an employee vs computer algorithm (orange) and monkey vs computer algorithm (cyan) for 150 trials/block as function of the inspection cost. The solid line indicates the Nash equilibrium. (C) Reward as function of the inspection cost for 150 trials/block. Coloring as in (B). pRL simulations are more similar to the experimental data than the TD simulations. (D) Average choice behavior as in (B) but for 5000 trials/block. The inspect rates for pRL vs pRL (TD vs TD) (dark (light) red circles) and pRL vs computer algorithm (purple squares) are shown too. The lines indicate the Nash equilibrium for the employee (diagonal) and the employer (horizontal). pRL behaves according to the Nash equilibrium, whereas TD does not. (E) Time course of the probability to shirk with inspection cost $i=0.7$ for pRL vs algorithm (blue line) and pRL vs pRL (TD vs TD) (dark (light) green line). For the latter the probability of the employer to inspect is shown too (dark (light) red line). pRL oscillates around the Nash equilibrium (drawn lines), whereas TD completely deviates from Nash. (F) Time course of the probability to shirk or inspect respectively with inspection cost $i=0.7$ for pRL vs pRL (green respectively red, solid) as in E, but shifted up for clarity and overlaid with the negative change in the shirk rate (green dashed) and the change in the inspect rate (red dashed) to show the counteractive behavior.

doi:10.1371/journal.pcbi.1002691.g003

costs $i \neq 0.5$, for instance, the softmax policy, which selects action a with probability $p_a \propto \exp(\beta Q_a)$, where Q_a is the value of action a and β a parameter (inverse temperature) regulating the amount of stochasticity in the decision making, does never reflect the Nash probabilities $p(\text{shirk}) = i$ and $p(\text{inspect}) = 0.5$, whatever the parameter β is (see Supporting Text S1).

In other learning tasks, animals and humans *transiently* match the choice probabilities with the average payoffs of the corresponding actions, $p_a \propto Q_a$ (for a discussion of probability matching see [36–38]). In these cases too, TD-learning with softmax will not be able to find the solutions neither, unless again all p- and Q-values are each the same, or alternatively, the choice policy is redefined (to $p_a \propto Q_a$, see [39]). For other examples where TD-learning fails in each of the two ways of either learning the values or inferring the choice probabilities see [7].

Not all covariance-rules lead to Nash equilibria. Covariance-based learning rules change the synaptic strengths $\mathbf{w}$ according to the covariance between reward prediction error and some measure of neuronal activity N, $\langle \Delta \mathbf{w} \rangle \propto \langle \text{Cov}(R - \langle R \rangle, N) \rangle$, see e.g. [37], where $\langle \cdot \rangle$ denotes expectation. pRL which follows the stochastic gradient of the expected reward, $\langle \Delta \mathbf{w} \rangle \propto \nabla \langle R \rangle$, is a special instance of a covariance rule where the quantity N corresponds to the eligibility trace E (Eq. 1). Steady-states of covariance rules satisfy Herrnstein's *matching law* [37,40]. This law states that the number of times an action is chosen is proportional to the reward accumulated from choosing that action. Formally, $n_a = \alpha R_a^{acc}$, where n_a is the number of times an action a is chosen, α is the action-independent proportionality constant, and R_a^{acc} is the reward accumulated by action a across its n_a choices.

Pure strategies, where only a single action is chosen, trivially satisfy the matching property since for non-chosen actions both n_a and R_a^{acc} vanish. In contrast, stochastic strategies only satisfy matching in the case that the selected options provide the same average reward $R_a^{acc}/n_a\ (=1/\alpha)$. The mixed (and trivially the pure) Nash equilibrium represents a special case of matching. If in a two-player game, for instance, player 2 adopts a mixed Nash strategy then, by definition, player 1 receives the same average reward $(1/\alpha)$ from any action (see also [38]).

Both the steady-state of a covariance rule and the Nash equilibrium imply matching. But a steady-state of the covariance rule does not necessarily need to be a Nash equilibrium. This can be seen by generalizing the classical covariance rule [37] to population learning, and applying this rule to the inspector game. To do so we replaced the eligibility trace E in the pRL rule (Eq. 1) by the deviation of the neuronal response c from its mean, $c - \bar{c}$.

The emerging population covariance (pCOV) learning rule,

$$\Delta w = \eta \,\mathrm{Dec}\, c (R-\bar{R})(c-\bar{c}) = \mathrm{Rew}\,\mathrm{Dec}\, c (c-\bar{c})\,, \qquad (2)$$

with positive learning rate η and estimates $\bar{R}$ (see Eq. 5 in Methods) and $\bar{c}$ for $\langle R\rangle$ and $\langle c\rangle$ respectively, does not follow the reward gradient and does not reliably converge to the unique mixed Nash equilibrium of the inspector game (Fig. 4). To keep simulation time short we did not use spiking neurons but merely binary neurons that produce output $c=\pm 1$ with probability $P_w(c)=\dfrac{1}{1+\exp(-c\mathbf{wx})}$ for a given binary input pattern $\mathbf{x}\in\{-1,1\}^{80}$. This yields an expected neural response $\bar{c}=\sum_c cP_w(c)=\tanh(\mathbf{wx}/2)$. Notice that $\mathrm{Rew}\,\mathrm{Dec}\, c$ can be considered as the personalized reward for the specific neuron in consideration: if the sign of Dec and c coincide, the population reward signal Rew elicited in response to the population decision can be taken as a personal reward signal for that neuron; otherwise the neuron's personal reward has reversed sign of Rew. Personalizing this way the neuronal rewards within the population solves the spatial credit-assignment problem and boosts learning [22]. Fig. 4 (A–D) shows the results for the pCOV rule applied to the inspector game, once with only the employee playing according to pCOV against an employer playing according to the algorithm presented in [11], and once for pCOV versus pCOV. Only in a fraction of the simulated runs is the mixed Nash equilibrium reached, while in the other runs, a deterministic (non-Nash) strategy pair emerges.

As check for robustness we performed further simulations showing that these negative results hold true also for other (non-gradient) covariance rules. We considered the version where the mean $\bar{c}$ is not calculated analytically but determined as a running average (as done for $\bar{R}$), and where the neuronal activity in the covariance rule is equal to the binary output, $N=c$, without mean subtraction (yielding the simple update rule $\Delta w=\mathrm{Rew}\,\mathrm{Dec}$). Moreover, considering only a single neuron and taking its response c as the behavioral decision did not qualitatively change the results, demonstrating that the failure is not due to the population framework (data not shown).

In [41] the author further elaborates on the relationship of covariance based rules to the Replicator dynamics. The latter is described by $\dfrac{\partial p_a}{\partial t}=\bar{\eta} p_a(\langle R|D=a\rangle-\langle R\rangle)$, where $\bar{\eta}>0$ is the effective learning rate and $\langle R|D=a\rangle$ is the average reward for choosing action a. The effective learning rate $\bar{\eta}$ depends on the details of the decision making network and is given in Eq.(14) of [41]. If synaptic changes are driven by the covariance of reward and neural activity, then according to the average velocity approximation, learning behavior is described by the differential equation above, but the effective learning rate $\bar{\eta}$ is not guaranteed to be positive. Indeed, for the binary neurons one gets $\bar{\eta}=\eta\sum_i \dfrac{x_i}{1+\cosh(\mathbf{wx})}$, which can be negative due to negative components x_i. Hence, the convergence statements in [41] do not apply to our decision making network, and there is no contradiction with the finding that the covariance rule fails to reproduce the mixed Nash equilibrium. Nevertheless, in the special case of a $0/1$ coding of the inputs x_i such that the effective learning rate becomes positive, and the specific covariance rule $\Delta w=\eta(R-\bar{R})(c-\bar{c})$ with a single postsynaptic neuron ($N=1$), we can also fit the human data (for 150 learning trials) and obtain the oscillations around the Nash equilibrium (for 5000 trials).

Whereas we do not consider the neural mechanisms supporting the readout of the decision, such a mechanism has been studied in the context of matching [42]. There, the probability for decision making is well described by a logistic function, which is also our choice for $P(D|A)$. For an increasing amount of stochasticity in the decision making they report an increasing deviation from matching towards more uniform choice probabilities. For choice probabilities larger than $1/2$ this leads to lower choice probabilities than predicted by the matching law, a phenomenon called undermatching. We also varied the amount of stochasticity by changing the slope of $P(D|A)$ as a function of A. We find that increasing and decreasing the slope by a factor of two still robustly leads to matching for the considered inspection costs, but when decreasing it by a factor of ten we also observe undermatching. Due to the stochasticity in the decision making the range of choice probabilities our network can represent is limited. With increasing the amount of stochasticity the range becomes smaller and extreme probabilities close to 0 or 1 predicted by the matching law cannot be represented. Instead, choice probabilities lie closer to uniform randomness, i.e. undermatching occurs.

pRL fits data and Nash better than basic reinforcement models. The Replicator equation is widely used in evolutionary game theory and provides a good phenomenological description of choice behavior in many repeated-choice experiments. A different phenomenological description has been suggested in [17]. Starting from Luce's basic reinforcement learning (RE1, see also [43]), the authors adapt this rule to take account of a generalization and recency effect (RE3, Methods). They show that both the basic and extended reinforcement learning reproduces the behavior of humans in many different games. However, we find that these rules poorly match human behavior for the inspector game. In fact, for 150 trials/block both models of Erev and Roth fit the experimental data significantly worse than pRL (Fig. 4 E and F). After 5000 trials RE3 converged to a pure (non-Nash) strategy, and for RE1 the inspection rate diverged away from the Nash equilibrium of 0.5. This non-optimal equilibrium performance is consistent with the fact that RE1 and RE3 are not gradient procedures (see Supporting Text S3). Whether pRL fits the behavioral data of humans also better in the other games Erev and Roth considered remains to be tested. In any case, the models have to cope with the fact that humans show a variety of behaviors in two-player matrix games, although in many settings they eventually play according to Nash (for a discussion see [44]).

Discussion

We considered a population of spiking neurons which represent an adaptive agent in a dynamic environment including other adaptive agents. The agent's adaptation was implemented as population reinforcement learning algorithm (pRL) which was previously shown to perform stochastic gradient ascent in the reward for partially observable Markov decision processes (POMDPs) [7]. Here we showed with blackjack and the inspector game that pRL can also cope with a dynamic multi-agent environment and that the performance is comparable to human data in both these games. In fact, when two neuronal populations play against each other, they learn to behave according to the optimal (but unstable) Nash equilibrium. By definition, no further increase in an agent's expected payoff is possible in the Nash equilibrium by only changing its own strategy while the environment remains stationary. In these steady-state conditions – where the opponent's strategy is assumed to be stationary – pRL is proven to maximize the expected reward (Supporting Text S2). The simulations show that the equilibrium is indeed reached by two pRL agents playing against each other, with a pure (deterministic) Nash equilibrium in blackjack and a mixed

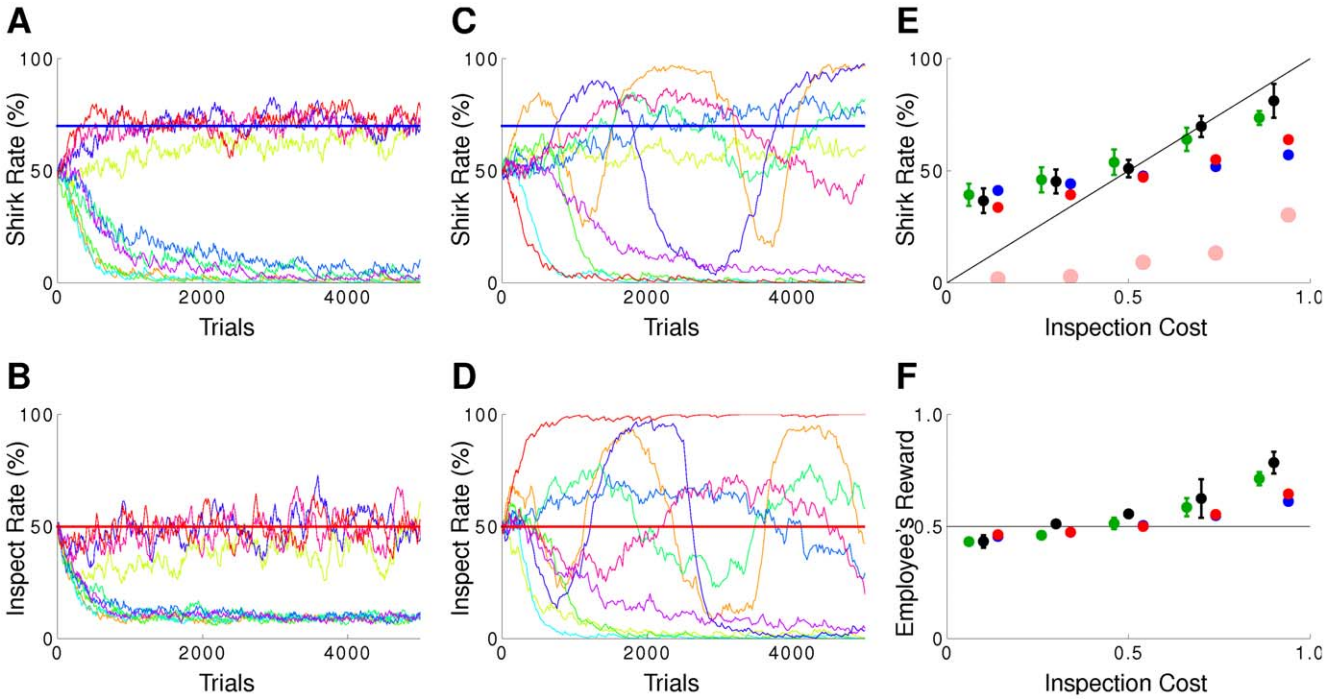

Figure 4. Covariance learning rules may lead to a mixed Nash equilibrium, but also to deterministic non-Nash strategies. pRL fits data better than basic reinforcement models. Time course of the probability to shirk (A,C) and inspect (B,D) with inspection cost $i=0.7$ for pCOV vs algorithm (A,B) and pCOV vs pCOV (C,D). In each panel the horizontal lines depict the Nash equilibrium, and for 10 simulation runs inspection and shirk rates are shown (same color in (A,B) and (C,D), respectively, correspond to the same run). Only a small fraction of all runs converge or oscillate around the Nash equilibrium, while the other runs result in a deterministic strategy pair. The initial distribution of synaptic weights w was Gauss with mean 0 and standard deviation 0.02. The learning rate was set to $\eta=0.004$, but $\eta=2\cdot10^{-7}$ did not change the proportion of runs converging to the pure strategy. (E) Average choice behavior of pRL vs pRL (green), RE1 vs RE1 (blue), RE3 vs RE3 (red) and human vs human (black) for 150 trials/block as function of the inspection cost. The light red circles show the average choice behavior for RE3 vs RE3 and 5000 trials/block. Individual runs converged to a pure strategy, hence the shown averages over 200 runs reflect the percentage of runs converging to a pure shirk strategy. (F) Reward as function of the inspection cost for 150 trials/block. Coloring as in (E). The solid lines indicate the Nash equilibrium.
doi:10.1371/journal.pcbi.1002691.g004

(stochastic) Nash equilibrium in the inspector game. As predicted by the theory [34,35], the strategies oscillated around the mixed Nash equilibrium when both players used the same gradient algorithm based on the others stationarity assumption, i.e. when one network played against another both using pRL (with a small learning rate). Averaging over long enough time windows, i.e. long compared to the oscillation period, yields the Nash equilibrium values. However, when implementing only the employee by a gradient pRL network and the employer by a non-gradient computer algorithm [11], the two players do not play exactly equally well. In this case no oscillations occurred and both converged to and stayed at the optimal Nash equilibrium.

For mathematical clarity we presented the spike-based pRL for an episodic learning scenario. But a biologically plausible implementation of a fully online scheme is also possible: to avoid an explicit separation of stimuli in time, the rectangular window function used to temporally integrate the eligibility trace (Eq. 8 in Methods) can be replaced by an exponentially decaying window function to get a low-pass filtered eligibility trace, and concentrations of neuromodulators can be used to encode feedback about the population decision and the global reward signal (e.g. acetylcholine or dopamine) [22]. We considered reward delivery immediately after stimulus presentation, but reward could also be substantially delayed when considering a further eligibility trace incorporating the population decision [7]. Moreover, since learning in general speeds up with population size (up to 1-shot learning for stimulus-response associations [31]) we expect that the convergence for pRL towards the Nash equilibrium can be much faster than in our example where parameters were fit to reproduce human data.

The mixed Nash equilibrium represents a special case of Herrnstein's matching law [40], according to which the number of times an action is chosen is proportional to the reward accumulated from choosing that action. This is true both for the pure and mixed Nash optimum. In the special case that the current reward only depends on the current action, but not on past actions, reward maximization always implies matching. (In fact, if one action would yield a higher (average) payoff per choice, then this action must be chosen with probability 1 to maximize expected reward, and matching ($n_a=\alpha R_a^{acc}$) is trivially satisfied (since for the non-chosen action $n_a=R_a^{acc}=0$). If both actions yield the same payoff R_a per choice ($=R_a^{acc}/n_a=1/\alpha$), then matching is again trivially satisfied.) In turn, a reward-based learning rule which only empirically maximizes reward in this case leads to only an approximated matching [42]. Choice probabilities which maximize the expected reward are trivially also fixed points of any learning rule defined by the covariance between reward and neuronal activity. (In fact, at the reward maximum there is no change in neuronal activity which, in average, would lead to an increase (and in the opposite direction to a decrease) of the expected reward, and hence the covariance between activity and reward must vanish.) The other direction, again, is not true: a covariance-based rule does not necessarily lead to reward maximization or a Nash equilibrium [37,38]. Indeed, our

simulations of the inspector game with the canonical covariance-based plasticity rules show that these rules do not necessarily lead to the mixed Nash equilibrium, but instead can result in deterministic (non-Nash) strategies. Similarly, basic reinforcement rules studied in the context of economics and human decision making [17] are neither compatible with the mixed Nash equilibrium for the inspector game.

The performance of spike-based pRL is also superior to TD-learning [2] which is often discussed in the neuro-economical context [1]. With the parameter values for which TD-learners came closest to human data (although without matching them as closely as pRL), the mixed Nash equilibrium in the inspector game was not reached within the long learning times. Instead, TD-learner first adopted a deterministic strategy, transiently switched their behavior, and swapped back to the same deterministic strategy. We attributed this mismatch to a general failing of TD-learning in correctly mapping action values to choice probabilities in probabilistic decision making tasks. TD-learning with the softmax choice policy, in particular, fails when matching of choice probabilities with average payoff is required [40].

Different generalizations have been considered to approach the shortcomings of algorithms in socio-economic games. TD-learning has been extended to not only assign values to its own decisions, but to pairs of own and opponent decisions. This enables the learning of minimax strategies where reward is maximized for the worst of the opponents actions [45]. While for zero-sum games minimax may realize a mixed Nash equilibrium, it results in a deterministic strategy in the inspector game: minimizing the maximal loss implies for the employee to always work (to prevent being caught while shirking), and for the employer to always inspect (to prevent undetected shirking). Another approach is to separately learn its own and the opponents action values and then calculate the Nash equilibrium [46], but such explicit calculations do not seem to be the typical human behavior in socio-economic interactions. Instead, it is tempting to consider pRL with long eligibility traces which, as it performs policy gradient in POMDPs [7], should find cooperative strategies with, on average, higher than Nash payoffs for all agents. For the inspector game such a co-operative strategy is that the employer should let the employee sporadically shirk (say with probability ϵ) without inspection, but with the common agreement that shirking will not prevail (leading to average payoffs $\frac{1}{2}(1+\epsilon)$ and $2(1-\epsilon)$ for the employee and employer, respectively).

Although under the specific experimental conditions of the inspector game humans did not show cooperation, they often do so in other game-theoretic paradigms, as e.g. in the prisoner's dilemma, and hence deviate from the Nash equilibrium (for a review see [47]). It remains a challenge for future modeling work to capture such cooperative behavior. Likely, this will involve modeling the prediction of other player's reactions in response to ones own actions, as considered in the theory of mind [48] and as being a hallmark of successful socio-economic behavior.

Given the difficulties of modeling genuine social behavior, and the difficulties humans effectively have in stacked reflexive reasoning, the assumption of the opponent's stationarity considered here appears as a reasonable approximation for decision making even in complex situations. In view of its success in matching behavioral and theoretical data we may ask how far human decision making is in fact determined by cognitive reasoning, or whether decisions should rather be attributed to automated neuronal processes steered e.g. by pRL (which can also encompass input from a cognitive module as it is suggested for the production rules in the ACT-R theory, [18]). In fact, daily experience tells us that decisions are often more appropriate when we listen to our gut feeling, while we tend to merely add justifications post-hoc. Or put in Schopenhauer's words, "that in spite of all his resolutions and reflections he does not change his conduct" [49].

Methods

Model details

Focusing on one neuron we denote by $\mathbf{X}$ its input, which is a spike pattern made up of M spike trains, and by Y its output spike train. The membrane potential can be written as

$$u(t)=u_0+\sum_{i=1}^{M} w_i \sum_{s\in X_i} \epsilon(t-s)-\sum_{s\in Y} \kappa(t-s). \tag{3}$$

The postsynaptic kernel $\epsilon(t)$ and the reset kernel $\kappa(t)$ vanish for $t\leq 0$. For $t>0$ they are given by

$$\epsilon(t)=\frac{1}{\tau_M-\tau_S}\left(e^{-t/\tau_M}-e^{-t/\tau_S}\right) \text{ and } \kappa(t)=\frac{1}{\tau_M}e^{-t/\tau_M}.$$

For the resting potential we use $u_0=-1$ (arbitrary units). Further, $\tau_M=10\,\text{ms}$ is used for the membrane time constant and $\tau_S=1.4\,\text{ms}$ for the synaptic time constant. Action potential generation is controlled by an instantaneous firing rate $\phi(u)$ which increases with the membrane potential. So, at each point t in time, the neuron fires with probability $\phi(u(t))\delta t$ where δt represents an infinitesimal time window (we use $\delta t=0.2\,\text{ms}$ in the simulations). Our firing rate function is

$$\phi(u)=ke^{\beta u},$$

with $k=0.01$ and $\beta=5$ (parameter values taken from [29], see also [19]).

We consider a population of $N=100$ neurons and an input layer of size $M=80$ for each player that is represented by a neural net. We assume that each population neuron synapses onto a site in the input layer with probability of 80%, leading to many shared input spike trains between the neurons. The population response is read out by the decision making unit based on a spike/no-spike code. We introduce the coding function $c(Y^v)$, with $c(Y^v)=-1$, if neuron v does not spike, otherwise $c(Y^v)=1$. The population activity A being read out by the decision making unit is:

$$A(\mathbf{Y})=\frac{1}{\sqrt{N}}\sum_v c(Y^v).$$

Note that such a formal summation could be implemented in terms of a neuronal integrator (forming a 'line attractor') which continuously integrates excitatory and inhibitory input and keeps the neuronal activity at a constant level in the absence of input [50]. Using this activity readout, the behavioral decision $D=\pm 1$ is made probabilistically, with likelihood $P(D=1|A)$ given by the logistic function

$$P(D=1|A)=\frac{1}{1+\exp(-A)}, \tag{4}$$

and $P(D=-1|A)$ being the counter probability. The normalization of the activity A with $\sqrt{N}$ ensures that $\mathrm{Var}(A)=O(1)$, thus being of same order as the noise in the decision readout.

We now describe the terms, modulating synaptic plasticity in Eq. (1). The reward feedback Rew encodes the reward prediction error, as observed in experiments [20],

$$\mathrm{Rew} = \eta(R - \bar{R}). \tag{5}$$

Here $\bar{R}$ is a running mean estimate of the expected reward, $\bar{R} \leftarrow (1-\lambda)\bar{R} + \lambda R$, where we set $\lambda = 0.1$. The parameter η is the positive learning rate which, for notational convenience, we absorb into the reward signal. In all pRL simulations we used the value $\eta = 400$. Both values λ and η (rounded) were chosen to minimize the Mean Squared Error (MSE) between the average model and human data ($\mathrm{MSE} = 0.0027$ for the shirk rate and $\mathrm{MSE} = 0.0028$ for the employee's reward in the inspector game). All other parameter values were taken from [7].

The decision feedback Dec is given by

$$\mathrm{Dec} = \frac{D}{1 + \exp(DA)}, \tag{6}$$

which is the derivative of $\log P(D|A)$, see Eq. (4), with respect to A; so decision feedback measures how sensitive the decision is to changes in activity.

As shown in [29], the probability density, $P_w(Y)$, that a neuron actually produces the output spike train Y in response to the stimulus $\mathbf{X}$ during a decision period lasting from $t = 0$ to $t = T$ satisfies:

$$\ln P_w(Y) = \sum_{s \in Y} \ln \phi(u(s)) - \int_0^T \mathrm{d}t\, \phi(u(t)). \tag{7}$$

The derivative of $\ln P_w(Y)$ with respect to the strength of synapse i is known as characteristic eligibility in reinforcement learning [51]. For our choice of the firing rate function one obtains for the last term in (1)

$$E = \frac{\partial}{\partial w_i} \ln P_w(Y) = \int_0^T \mathrm{d}t \left(\sum_{s \in Y} \delta(t-s) - k\, e^{\beta u(t)} \right) \beta \sum_{s \in X_i} \epsilon(t-s). \tag{8}$$

In all the simulations initial values for the synaptic strength were picked from a Gaussian distribution with mean zero and standard deviation equal to 4, independently for each afferent and each neuron. In the Supporting Text S2 we show that the plasticity rule (1) composed of the factors (5, 6, 8) and the decision c follows the stochastic gradient of the expected reward.

TD-learning

For TD-learning we used the SARSA control algorithm [2] which estimates the values of state-action pairs (s,a). At each point in time, the value estimates $Q_a(s)$ are updated according to

$$Q_a(s) \leftarrow Q_a(s) + \alpha(R - Q_a(s)).$$

Here α is similar to a learning rate and has values between 0 and 1. R is the reward immediately obtained after performing action a. In the case of blackjack it is defined as zero if the game is not over and the player chooses to draw another card, otherwise it is determined by the payoffs of the considered game. When in state s, the next action D is chosen using softmax, i.e. according to the probability $P(D = a) \propto \exp(\beta Q_a(s))$. In all simulations we used the rounded values $\alpha = 0.004$ and $\beta = 50$ as they minimized the MSE between averaged model and human data ($\mathrm{MSE} = 0.0048$ for the shirk rate and $\mathrm{MSE} = 0.0110$ for the employee's reward in the inspector game). Note that in both TD-learning and pRL we adapted the same number of free parameters (TD: α and β; pRL: λ and η), making it possible to directly compare the quality of the fit.

Basic reinforcement models

In both Roth-Erev models [17] the probabilistic choice rule is parametrized using propensities q_i. The probability p_k that a specific player (who's index is omitted) plays his kth pure strategy is $p_k = \frac{q_k}{\sum_l q_l}$.

RE1 model. The propensity of the chosen action k is incremented by the received reward R_k, $q_i \leftarrow q_i + R_k \delta_{ik}$, where δ_{ik} denotes the Kronecker delta. The initial propensities are set to $q_i = s\,X$, where X is the average reward for that player under uniformly distributed p_k's, and the strength parameter s is the one parameter that is optimized.

RE3 model. In addition to the strength parameter s, the generalization and forgetting parameters ϵ and ϕ are introduced. The propensities are updated according to

$$q_i \leftarrow (1-\phi) q_i + R_k(1-\epsilon)\delta_{ik} + R_k\,\epsilon(1-\delta_{ik}).$$

The parameters were chosen to minimize the mean squared error (MSE) between the average model and human data ($\mathrm{MSE} = 0.0188$ (RE1) and 0.0116 (RE3) for the shirk rate and $\mathrm{MSE} = 0.0081$ (RE1) and 0.0060 (RE3) for the employee's reward in the inspector game).

Blackjack details

In blackjack we assume an infinite number of card decks. Independently of the history, the drawing probability therefore remains constant, with a probability to draw a card with value 10 being $\frac{4}{13}$, and the probability to draw any other value from 2 to 11 being $\frac{1}{13}$. For a strategy determined by the stopping value s we calculated analytically the probability distribution of hand values $P_s(v)$ after drawing the last card. The drawing process is iterated for those hand values that are smaller than s until there is only probability mass on hand values greater than or equal to s. Because the lowest card value on the desk remains always 2, drawing i times in a row yields a lowest possible hand value of $2i$. Hence up to $\lceil s/2 \rceil$ cards are drawn in order to obtain a hand value v greater or equal to s. Let us denote the value of the ith card by v_i and its probability distribution by $Q(v_i)$,

$$Q(v_i) = \begin{cases} \frac{1}{13} & \text{for} \quad v_i \in \{2,3,\ldots,9,11\} \\ \frac{4}{13} & \text{for} \quad v_i = 10 \\ 0 & \text{else.} \end{cases}$$

To obtain the probability distribution $P_s(v)$ we sum up the probabilities of all possible combinations to draw $k \leq \lceil s/2 \rceil$ cards that yield hand value v, with the condition that the sum of the first $k-1$ drawn cards is smaller than s, such that a kth card is actually drawn under the stopping strategy.

Table 3. Probability distribution of hand values $P_s(v)$ after drawing the last card.

v	<15	15	16	17	18	19	20	21	>21
$P_{15}(v)$	0	0.1206	0.1247	0.1194	0.1138	0.1078	0.1546	0.0944	0.1648
$P_{16}(v)$	0	0	0.1247	0.1287	0.1231	0.1170	0.1638	0.1036	0.2390

doi:10.1371/journal.pcbi.1002691.t003

$$P_s(v)=\begin{cases}0 & \text{for} \quad v<s\\ \sum_{k=1}^{\lceil s/2\rceil}\sum_{v_1,\ldots,v_k=2}^{11} Q(v_1)\cdots Q(v_k)\,\mathrm{II} & \\ \left(\sum_{i=1}^{k} v_i=v\right)\mathrm{II}\left(\sum_{\mathrm{i}=1}^{\mathrm{k}-1}\mathrm{v_i}<\mathrm{s}\right) & \text{else}\end{cases}$$

where II is the indicator function which is one if its argument is true and zero else. The product of the $Q(v_i)$ is the joint probability that the first card has value v_1, the second v_2 and so on. The first indicator function ensures that all k drawn cards sum up to v, the second that k cards are drawn, i.e. the sum of the first $k-1$ cards has to be smaller than the stopping value s, because otherwise no further card would be drawn. For instance in the case of $s=15$ and 16 one obtains the distributions in Table 3.

Denoting the hand value of the gambler by v_1 and that of the croupier by v_2 the payoff of the bank is

$$R_2(v_1,v_2)=\begin{cases}+1 & \text{for} \quad v_1\le v_2\le 21 \quad \text{or} \quad v_1>21\\ -1 & \text{else}\end{cases}$$

Averaging of R_2 with respect to the joint distribution $P_{s_1,s_2}(v_1,v_2)=P_{s_1}(v_1)P_{s_2}(v_2)$ yields the entry in the average payoff matrix Table 1 for the strategy pair (s_1,s_2). For instance for $(s_1=15,s_2=16)$, $\langle R_2(v_1,v_2)\rangle_{P_{15,16}(v_1,v_2)}=\sum_{v1,v2}P_{15}(v_1)P_{16}(v_2)R_2(v_1,v_2)=0.1555$.

We defined the drawing probabilities in Fig. 2 for a hand value at a certain game number g as the frequency with which another card has been drawn upon the last 20 presentations prior to g of the corresponding stimulus. The evolution of the average reward $\hat{R}_g$ in time in Fig. 2C are the low pass filtered reward sequences, $\hat{R}_g=(1-\lambda)\hat{R}_{g-1}+\lambda R_g$ where R_g is the reward in the g-th game and $\lambda=0.002$ was used. The initial value $\hat{R}_0$ was calculated assuming a random 50% choice behavior prior to learning.

The initial weights mimicking the prior strategy of instructed humans were obtained by training our network to make a decision with a certain probability. This is possible by adapting pRL to perform regression (as will be published elsewhere).

Inspector game details

The evolution of the rates in time in Fig. 3E are the low pass filtered decision sequences, e.g. $p_t(\text{shirk})=(1-\lambda)p_{t-1}(\text{shirk})+\lambda d_t$ where $d_t=1$ if the employee shirks in trial t, otherwise $d_t=0$. We used a value of $\lambda=0.02$ and assumed again an initial random 50% choice behavior. The rate change in Fig. 3F was determine by binning the obtained time course of the rate into bins of width 20, calculating the mean of each bin, and the differences between succeeding bins. The result was further low pass filtered once more with an exponential running mean ($\lambda=0.1$) to reduce the noise.

Supporting Information

Text S1 We present further results for temporal-difference learning and elaborate on its failure to learn mixed Nash equilibria.
(PDF)

Text S2 We show how the plasticity rule presented in the main text is based on a gradient ascent procedure maximizing the average reward.
(PDF)

Text S3 We demonstrate that the heuristic rules of Erev and Roth [17] are no gradient procedures.
(PDF)

Acknowledgments

We thank Robert Urbanczik for helpful discussions, Michael C. Dorris for providing the code of the computer algorithm used in [11], Johannes Hewig for the data of humans playing blackjack [10], and Yonatan Loewenstein for helpful feedback on the covariance rules.

Author Contributions

Conceived and designed the experiments: JF WS. Performed the experiments: JF. Analyzed the data: JF WS. Wrote the paper: JF WS.

References

1. Dayan P, Daw ND (2008) Decision theory, reinforcement learning, and the brain. Cogn Affect Behav Ne 8: 429–453.
2. Sutton RS, Barto AG (1998) Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press.
3. Seymour B, O'Doherty JP, Dayan P, Koltzenburg M, Jones AK, et al. (2004) Temporal difference models describe higher-order learning in humans. Nature 429: 664–7.
4. Potjans W, Morrison A, Diesmann M (2009) A spiking neural network model of an actor-aritic learning agent. Neural Comput 21: 301–339.
5. Howard RA (1960) Dynamic programming and Markov processes. Cambridge, MA: MIT Press.
6. Smallwood RD, Sondik EJ (1973) The optimal control of partially observable markov processes over a finite horizon. Oper Res 21: 1071–1088.
7. Friedrich J, Urbanczik R, SennW(2011) Spatio-temporal credit assignment in neuronal population learning. PLoS Comput Biol 7: e1002092.
8. Fudenberg D, Levine DK (1998) Theory of Learning in Games. Cambridge, MA: MIT Press.
9. Von Neumann J, Morgenstern O (1944) Theory of games and economic behavior. Princeton: Princeton University Press.
10. Hewig J, Trippe R, Hecht H, Coles GH, Holroyd CB, et al. (2007) Decision-making in blackjack: An electrophysiological analysis. Cereb Cortex 17: 865–877.
11. Dorris MC, Glimcher PW (2004) Activity in posterior parietal cortex is correlated with the relative subjective desirability of action. Neuron 44: 365–378.
12. Thorndike EL (1898) Animal Intelligence: An Experimental Study of the Associative Processes in Animals. Psychol Monogr 2: 321–330.
13. Rescorla R, Wagner A (1972) A theory of Pavlovian conditioning: variations in the effectiveness of reinforecement and nonreinforcement. In: Black AH, Prokasy WF, editors. Classical Conditioning II: current research and theory. New York: Appleton Century Crofts. pp. 64–99.

14. Dayan P, Long T (1998) Statistical models of conditioning. Adv Neural Inf Process Syst 10. pp. 117–123.
15. Fiete IR, Seung HS (2006) Gradient learning in spiking neural networks by dynamic perturbation of conductances. Phys Rev Lett 97: 048104.
16. Frémaux N, Sprekeler H, Gerstner W (2010) Functional requirements for reward-modulated spike- timing-dependent plasticity. J Neurosci 30: 13326–13337.
17. Erev I, Roth AE (1998) Predicting how people play games: reinforcement learning in experimental games with unique, mixed strategy equilibria. Amer Econ Rev 88: 848–881.
18. Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, et al. (2004) An integrated theory of the mind. Psychol Rev 111: 1036–1060.
19. Gerstner W, Kistler WM (2002) Spiking Neuron Models. Cambridge, UK: Cambridge University Press.
20. Schultz W, Dayan P, Montague PR (1997) A neural substrate of prediction and reward. Science 275: 1593–1599.
21. Wang XJ (2008) Decision making in recurrent neuronal circuits. Neuron 60: 215–234.
22. Urbanczik R, Senn W (2009) Reinforcement learning in populations of spiking neurons. Nat Neurosci 12: 250–252.
23. Izhikevich EM (2007) Solving the distal reward problem through linkage of STDP and dopamine signaling. Cereb Cortex 17: 2443–2452.
24. Averbeck B, Latham PE, Pouget A (2006) Neural correlations, population coding and computation. Nat Rev Neurosci 7: 358–3666.
25. Von Neumann J (1928) Zur Theorie der Gesellschaftsspiele. Math Ann 100: 295–320.
26. Hewig J, Trippe R, Hecht H, Coles GH, Holroyd CB, et al. (2008) An electrophysiological analysis of coaching in blackjack. Cortex 44: 1197–1205.
27. Avenhaus R, Von Stengel B, Zamir S (2002) Inspection games. In: Aumann RJ, Hart S, editors. Handbook of Game Theory with Economic Applications. pp. 1947–1987.
28. Glimcher PW, Camerer C, Fehr E, Poldrack R, editors (2008) Neuroeconomics: decision making and the brain. Amsterdam: Elsevier.
29. Pfister J, Toyoizumi T, Barber D, Gerstner W (2006) Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. Neural Comput 18: 1318–1348.
30. Florian RV (2007) Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. Neural Comput 19: 1468–1502.
31. Friedrich J, Urbanczik R, Senn W (2010) Learning spike-based population codes by reward and population feedback. Neural Comput 22: 1698–1717.
32. Seung HS (2003) Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. Neuron 40: 1063–1073.
33. Werfel J, Xie X, Seung HS (2005) Learning curves for stochastic gradient descent in linear feedfor-ward networks. Neural Comput 17: 2699–2718.
34. Crawford VP (1985) Learning behavior and mixed-strategy Nash equilibria. J Econ Behav Organ 6: 69–78.
35. Stahl DO (1988) On the instability of mixed-strategy Nash equilibria. J Econ Behav Organ 9: 59–69.
36. Shanks DR, Tunney RJ, McCarthy JD (2002) A re-examination of probability matching and ra-tional choice. J Behav Decis Making 15: 233–250.
37. Loewenstein Y, Seung HS (2006) Operant matching is a generic outcome of synaptic plasticity based on the covariance between reward and neural activity. Proc Natl Acad Sci U S A 103: 15224–15229.
38. Loewenstein Y, Prelec D, Seung HS (2009) Operant matching as a Nash equilibrium of an in-tertemporal game. Neural Comput 21: 2755–2773.
39. Sakai Y, Fukai T (2008) The actor-critic learning is behind the matching law: matching versus optimal behaviors. Neural Comput 20: 227–251.
40. Herrnstein RJ (1961) Relative and absolute strength of response as a function of frequency of reinforcement. J Exp Anal Behav 4: 267–272.
41. Loewenstein Y (2010) Synaptic theory of replicator-like melioration. Front Comput Neurosci 4: 17.
42. Soltani A, Wang XJ (2006) A biophysically based neural model of matching law behavior: melio-ration by stochastic synapses. J Neurosci 26: 3731–3744.
43. Luce R (1959) Individual Choice Behavior: A Theoretical Analysis. New York: Wiley.
44. Shanks DR, Tunney RJ, McCarthy JD (1998) Do people play Nash equilibrium? Lessons from evolutionary game theory. J Econ Lit 36: 1–28.
45. Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the International Conference on Machine Learning. San Francisco, CA. pp. 157–163.
46. Hu J, Wellman MP (2003) Nash Q-learning for general-sum stochastic games. J Mach Learn Res 4: 1039–1069.
47. Holt CA, Roth AE (2004) The Nash equilibrium: a perspective. Proc Natl Acad Sci U S A 101: 3999–4002.
48. Yoshida W, Dolan RJ, Friston KJ (2008) Game theory of mind. PLoS Comput Biol 4: e1000254.
49. Schopenhauer A (1890) The wisdom of life. London: S. Sonnenschein. 147 pp.
50. Seung HS (1996) How the brain keeps the eyes still. Proc Natl Acad Sci U S A 93: 13339–13344.
51. Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn 8: 229–256.

Spike-based decision learning of Nash equilibria in two-player games.
Text S1: Further results for temporal-difference learning

Johannes Friedrich[1], Walter Senn[1,*]

1 Department of Physiology and Center for Cognition, Learning and Memory, University of Bern, Bühlplatz 5, CH-3012 Bern, Switzerland

∗ **E-mail: senn@pyl.unibe.ch**

In this Supplementary Material we present further results for TD-learning and elaborate on its failure to learn mixed Nash equilibria.

TD successfully learns a pure Nash equilibrium

TD-learning of blackjack yields similar results as spike-based population reinforcement learning (pRL). In Fig. S1 results for both algorithms are shown. TD learns the optimal deterministic decisions, even in the non-Markovian case of two TD-learners playing against each other (Fig. S1A).

TD fails to learn a mixed Nash equilibrium

The resulting asymptotic choice behavior of TD-learning in the inspector game depends on the inverse temperature in the softmax action selection. The Nash equilibrium is never reached, though for one TD-learner playing against the algorithm one can find an inverse temperature where the behavior is at least close to Nash (Fig. S2A). For two TD-learners (Fig. S2B) we calculated the asymptotic behavior by demanding that the average value update equals zero, i.e. the Q-values equal the expected payoffs. Using softmax action selection and consulting the payoff table Tab.2 this yields the following system of

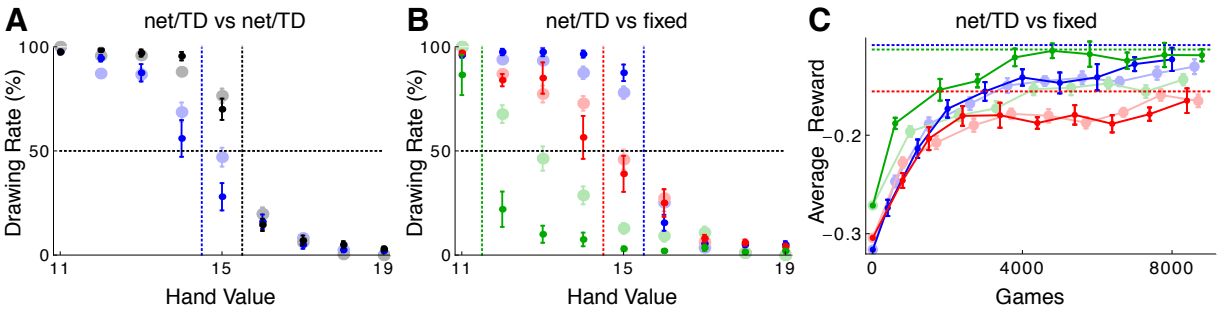

Figure S1. Playing blackjack yields similar results for pRL and TD. (A) Average strategy (±SEM) after 10 000 games where the gambler (blue) is a neural net (small dark circles) or TD-learner (large light circles) as well as the croupier (black). The vertical dotted lines left of $s_1 = 15$ and $s_2 = 16$ show the separation line of drawing/not drawing another card for the optimal Nash strategy pair. (B) Average strategy (±SEM) after 10 000 games for a neural net (small dark circles) or TD-learner (large light circles) as gambler playing against a croupier that follows a given strategy $s_2 = 15$ (blue), 16 (red) or 17 (green). The colored dotted lines left of $s_1 = 12, 15, 16$ show the separation line of drawing/not drawing another card for the optimal strategy given that the croupier stops drawing at $s_2 = 17, 16, 15$ (from left to right). (C) Average reward (±SEM) of the gambler for the scenario described in (B). The colored dotted lines show the maximal reachable average reward.

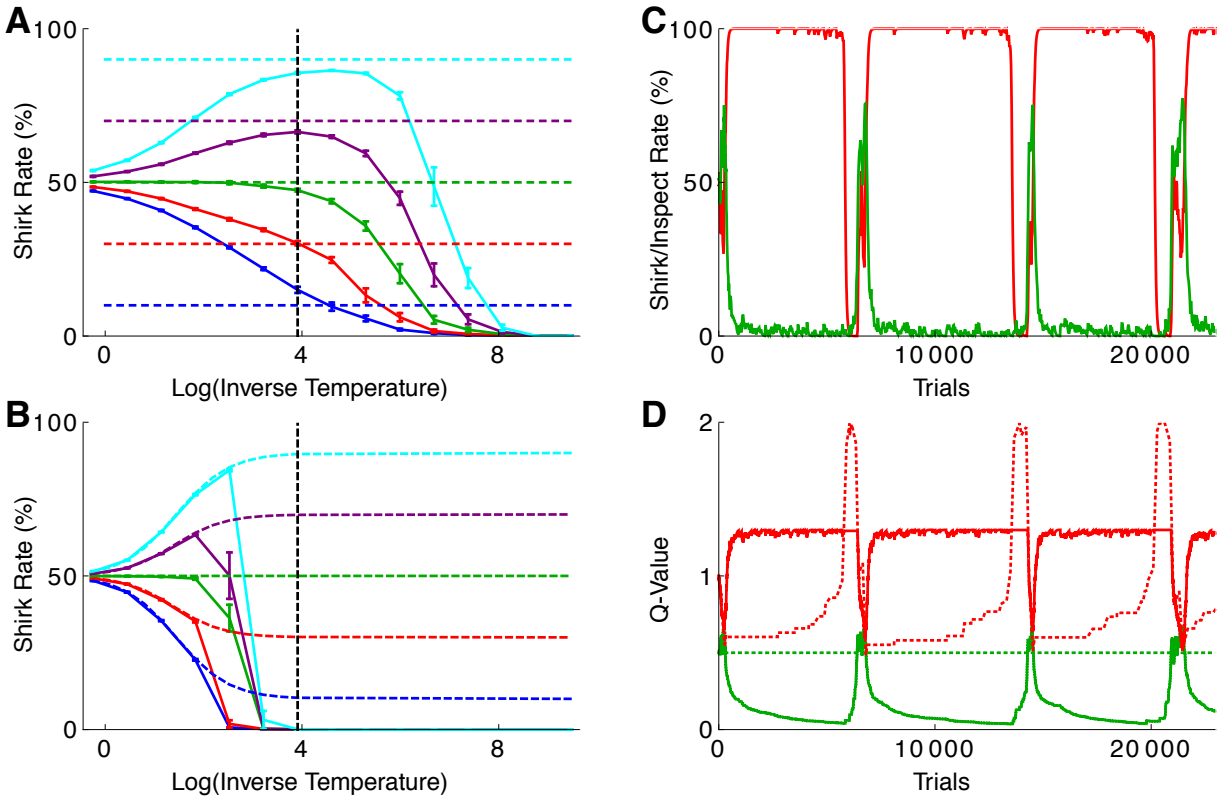

Figure S2. TD-learning in the inspector game strongly depends on the inverse temperature and shows oscillations. (A) Average choice behavior for TD vs computer algorithm over 5 000 trials as function of the inverse temperature β for inspection cost $i = 0.1$ (blue), 0.3 (red), 0.5 (green), 0.7 (purple) and 0.9 (cyan). The colored dashed lines indicate the Nash equilibrium, the black dashed line the value $\beta = 50$ chosen in Fig. 3 of the main text to best fit the experiments. (B) Average choice behavior for TD vs TD. The colored dashed lines show the solution of the equation system obtained when the average TD-update equals zero. (C) Single run shirk rate (green) and inspect rate (red) as function of time for TD vs TD. (D) Q-value of shirking (solid green), working (dashed green), inspecting (solid red) and not inspecting (dashed red) as function of time for the same single run as in (C). In (A) and (B) the same value $\alpha = 0.004$ as in the main text was used, whereas in (C) and (D) the parameters were set to $\alpha = 0.2$ and $\beta = 10$ in order to demonstrate the oscillatory behavior of TD-learning. With the values used in the main text the period of the oscillations would exceed reasonable simulation times.

equations:

$$p_s = \frac{e^{\beta Q_s}}{e^{\beta Q_s} + e^{\beta Q_w}} = \frac{1}{1 + e^{\beta(Q_w - Q_s)}} \tag{S1}$$

$$p_i = \frac{e^{\beta Q_i}}{e^{\beta Q_i} + e^{\beta Q_n}} = \frac{1}{1 + e^{\beta(Q_i - Q_n)}} \tag{S2}$$

$$Q_s = 1 - p_i \tag{S3}$$

$$Q_w = 0.5 \tag{S4}$$

$$Q_i = (2 - i)(1 - p_s) + (1 - i)p_s \tag{S5}$$

$$Q_n = 2(1 - p_s) \tag{S6}$$

where we introduced the notations $p_s = p(\text{shirk})$, $p_i = p(\text{inspect})$, $Q_s = Q(\text{shirk})$, $Q_w = Q(\text{work})$, $Q_i = Q(\text{inspect})$ and $Q_n = Q(\text{don't inspect})$. Given the inspection costs i and some inverse temperature β, one can plug in the Q-values (Eqs. (S3)–(S6)) into Eqs. (S1) and (S2) and solve for p_s and p_i. It turns out that, unless $i = 0.5$, the Nash equilibrium $p_s = i$ and $p_i = 0.5$ is never a solution of this system for any β. Hence, TD learning can never find the Nash equilibrium.

We numerically solved the above system for different i and β and plotted p_s as a function of β for different values of i (Fig. S2B). For increasing values of β the shirk probability p_s comes closer to the Nash value $p_s = i$. But for large β's slightly imprecise estimates of the Q-values will push the shirk and inspection probabilities to either 0 or 1, as can be seen from the very right of Eqs. (S1) and (S2). In numerical simulations this is expressed by the oscillations found in the action rates and the Q-values (Fig. S2C,D). There are long phases were the employee nearly always works and the employer inspects. Given that the employee shirks rarely, choosing to not inspect would actually yield a higher payoff for the employer, but the value of this action is still low and only updated in exploratory steps. Finally it surpasses the value of inspection and there is a phase were the employer does not inspect, which is realized after some exploratory steps by the employee who starts shirking. This leads to a drastic change in the employer's payoff and the employer's Q-values drop quickly to a low level. The employer inspects more forcing the employee to work. When the employee works again the payoff of the employer increases which he attributes to his inspection. The Q-value for inspection increases rapidly while the option to not inspect is rarely taken and its Q-value remains low, thus the cycle repeats. The period of the cycle is determined by how often exploratory steps are taken and due to the softmax action selection grows exponentially with the inverse temperature.

Spike-based decision learning of Nash equilibria in two-player games.
Text S2: Relating the plasticity rule to a gradient ascent procedure

Johannes Friedrich[1], Walter Senn[1,*]

1 Department of Physiology and Center for Cognition, Learning and Memory, University of Bern, Bühlplatz 5, CH-3012 Bern, Switzerland

∗ **E-mail: senn@pyl.unibe.ch**

Here we show how the plasticity rule presented in the main text is based on a gradient ascent procedure. First, a formula is derived for the gradient of the probability of taking a behavioral decision with respect to synaptic strength of the population neurons. Finally, we show that the plasticity rule presented in the main text performs stochastic gradient ascent in the expected reward. The derivation is analog to [7], but without delayed reward.

Gradient for the behavioral decision

Let $\mathbf{X}$ be the spike pattern presented to the population neurons and $\mathbf{W}$ the matrix of their synaptic strength. The probability $P_{\mathbf{W}}(D)$ of responding with decision D to the stimulus X is

$$P_{\mathbf{W}}(D) = \int \mathrm{d}\mathbf{Y}\, P(D|A(\mathbf{Y})) \prod_{\nu=1}^{N} P_{\mathbf{W}^\nu}(Y^\nu)\,, \tag{S7}$$

where we suppressed the conditioning on X. To lighten the notation further, we focus on calculating the gradient of $P_{\mathbf{W}}(D|X)$ only with respect to the strength of one of the synapses (the expressions for the other synapses being entirely analogous). Let w denote the strength of the first synapse of the first population neuron and let $Y = Y^1$ the postsynaptic spike train produced by this neuron. To isolate the contribution of the first neuron we decompose the activity $A(\mathbf{Y})$ as

$$A(\mathbf{Y}) = \tfrac{1}{\sqrt{N}} c(Y) + A^{\setminus}(Y^2, \ldots, Y^N) \quad \text{with } A^{\setminus} = \frac{1}{\sqrt{N}} \sum_{\nu=2}^{N} c(Y^\nu)\,.$$

Plugging this into (S7) we can calculate the derivative of $P_{\mathbf{W}}(D)$ with respect to the single weight w performed in the Supplementary Materials of [7]. Changing the matrix index of $P_{\mathbf{W}}(D)$ to w we obtain

$$\tfrac{\partial}{\partial w} P_w(D) = \int \mathrm{d}Y\, \mathrm{d}A^{\setminus} P_w(D, A^{\setminus}, Y)\; \tfrac{1}{\sqrt{N}} \left(\tfrac{\partial}{\partial A} \ln P(D|A)\right) c(Y) \tfrac{\partial}{\partial w} \ln P_w(Y)\,. \tag{S8}$$

Gradient of the expected reward

Since we consider immediate reward application without delay (unlike in [7]), reward does only depend on the decisions D_1 and D_2 of both opposing agents in the current trial. We assume that the first agent is a population of neurons, whereas the decision making process of the second agent remains unspecified,

leaving the formalism general. The derivative of the first agent's expected reward $\langle R_1 \rangle$ in that trial is

$$\begin{aligned} \tfrac{\partial}{\partial w}\langle R_1\rangle &= \tfrac{\partial}{\partial w}\sum_{D_1,D_2} P_w(D_1,D_2)R(D_1,D_2) \\ &= \sum_{D_1,D_2} P(D_2|D_1)R(D_1,D_2)\tfrac{\partial}{\partial w}P_w(D_1) \\ &= \sum_{D_1,D_2} P(D_2|D_1)\left(R(D_1,D_2)-\bar{R}\right)\tfrac{\partial}{\partial w}P_w(D_1)\,. \end{aligned} \tag{S9}$$

In the last line we subtracted a term equaling zero for a reward baseline $\bar{R}$ that is conditionally independent of the current decisions D_1 and D_2, given the weights and the stimulus [51]. The choice of an adaptive estimate $\bar{R}$ of upcoming reinforcement based on past experience is known as reinforcement comparison [2]. The common approach we follow to compute $\bar{R}$ is to use the exponential averaging scheme. Formally, we assume that the probability distribution of the second agent's decisions conditioned on D_1, $P(D_2|D_1)$, is stationary. Plugging (S8) into (S9) yields

$$\begin{aligned} \tfrac{\partial}{\partial w}\langle R\rangle = \sum_{D_1,D_2}\int \mathrm{d}Y\,\mathrm{d}A^{\setminus}\, P_{w,t}(D_2,D_1,A^{\setminus},Y) \\ \times\left(R(D_1,D_2)-\bar{R}\right)\tfrac{1}{\sqrt{N}}\left(\tfrac{\partial}{\partial A}\ln P(D_1|A)\right)c(Y)\tfrac{\partial}{\partial w}\ln P_w(Y) \end{aligned} \tag{S10}$$

The first line is just the averaging operator. We can now compare the terms in the second line to the weight update (1),

$$\Delta w = \mathrm{Rew}\,\mathrm{Dec}\,c\,E\,, \tag{S11}$$

proposed in the main text. The first term corresponds to the reward signal $\mathrm{Rew} = \eta(R-\bar{R})$ given by Eq.(5). The derivative in the second term yields for the logistic function $P(D|A) = 1/(1+\exp(-DA))$ in Eq.(4) the decision feedback $\mathrm{Dec} = D/(1+\exp(DA))$ given by Eq.(6). The last term has already been introduced as eligibility trace E in Eq.(8). If we choose a small learning rate, the average over the decisions D_1, D_2, the postsynaptic spike train Y and the activity of the other neurons $A^{\setminus}$ can be replaced by a time average obtained by sampling these quantities. The corresponding online learning rule (1,S11) therefore results from dropping the averaging. While the transition from batch to online learning requires a small learning rate for the neuronal population even in a stationary environment, here further the learning rate of the second agent has to be small too, i.e. $P(D_2|D_1)$ changes slowly in time. On the other side the learning rate should not be too small in order to be able to react to changes in the dynamic environment.

Spike-based decision learning of Nash equilibria in two-player games.
Text S3: The Roth-Erev models are no gradient procedures

Johannes Friedrich[1], Walter Senn[1,*]

1 Department of Physiology and Center for Cognition, Learning and Memory, University of Bern, Bühlplatz 5, CH-3012 Bern, Switzerland

∗ **E-mail: senn@pyl.unibe.ch**

In this Supplementary Material we show that the Roth-Erev models [17] do not update the propensities in the gradient direction of the reward. For convenience we restate the update of the propensities for the 3-parameter model of Erev an Roth (ER3),

$$q_i \leftarrow (1-\phi)q_i + R_k(1-\epsilon)\delta_{ik} + R_k\,\epsilon\,(1-\delta_{ik})\,. \tag{S12}$$

Remember that the choice probabilities are set to $p_i = q_i/\sum_l q_l$. The one parameter model (ER1) is just a special case thereof with $\epsilon = \phi = 0$.

In a stochastic gradient procedure the average propensity change is proportional to the gradient of the expected reward. The latter is, suppressing the index n of the player, $\frac{\partial}{\partial q_i}\langle R\rangle = \sum_k p_k R_k \frac{\partial}{\partial q_i}\ln p_k$ where the last term evaluates to $\frac{\partial}{\partial q_i}\ln p_k = \frac{\delta_{ik}}{q_k} - \frac{1}{\sum_l q_l}$. The ensuing update rule is

$$q_i \leftarrow q_i + \eta\, R_k \left(\frac{\delta_{ik}}{q_k} - \frac{1}{\sum_l q_l}\right)\,, \tag{S13}$$

where the positive parameter η is the learning rate. The update differs from the update of RE3 (S12) for any choice of parameters.

To conclude already that RE3 is not a policy gradient procedure would be one step too fast. There are many different estimates for the reward gradient, $R_k(\frac{\delta_{ik}}{q_k} - \frac{1}{\sum_l q_l})$ is just one of them and maybe RE3 uses another one. We have to consider whether the *average* updates are equal. For the gradient procedure we obtain from averaging across the choice options $k = 1, 2$,

$$\langle \Delta q_i^{grad}\rangle = \eta\frac{\partial}{\partial q_i}\langle R\rangle = \eta\left(\frac{p_i R_i}{q_i} - \frac{\langle R\rangle}{\sum_l q_l}\right) = \frac{\eta}{\sum_l q_l}\left(R_i(1-p_i) - p_j R_j\right)\,, \tag{S14}$$

where i is one and j the other option. In contrast, for RE3 we obtain

$$\langle \Delta q_i^{RE}\rangle = -\phi q_i + p_i R_i(1-\epsilon) + p_j R_j \epsilon \tag{S15}$$

The average propensity update $\langle \Delta q_i^{RE}\rangle$ (S15) is never equal to $\langle \Delta q_i^{grad}\rangle$ (S14) for any parameter setting, hence the rule does not perform (stochastic) gradient ascent in the expected reward.