Cortical prediction markets

David Balduzzi
ETH Zürich, Switzerland
david.balduzzi@inf.ethz.ch

ABSTRACT
We investigate cortical learning from the perspective of mechanism design. First, we show that discretizing standard models of neurons and synaptic plasticity leads to rational agents maximizing simple scoring rules. Second, our main result is that the scoring rules are proper, implying that neurons faithfully encode expected utilities in their synaptic weights and encode high-scoring outcomes in their spikes. Third, with this foundation in hand, we propose a biologically plausible mechanism whereby neurons backpropagate incentives which allows them to optimize their usefulness to the rest of cortex. Finally, experiments show that networks that backpropagate incentives can learn simple tasks.

Categories and Subject Descriptors
[Theory of computation]: Multi-agent learning; [Theory of computation]: Algorithmic mechanism design

General Terms
Algorithms, Design, Theory

Keywords
incentives for cooperation, multiagent learning, biologically-inspired approaches, prediction markets

1. INTRODUCTION
How does the brain encode information about the environment into its structure [24]? Inspired by recent work in prediction markets, this paper investigates cortical learning and the neural code from the perspective of mechanism design [13,14,17]. To the best of our knowledge it is the first paper to do so.

We start in §2 by modeling neurons as rational agents: that is, agents whose sole aim is to maximize the expected value of an objective function. To do so, we draw on a recent paper showing that discretizing standard models of neuronal dynamics [13] and learning [23] yields a threshold neuron with an online update rule that optimizes a simple objective [3]. By maximizing their objective function, neurons seek to optimally trade off rewards, depending on neuromodulatory signals such as dopamine, with costs, depending on resources expended on synaptic connections [6,7].

However, it is not enough that neurons optimize locally. They should collectively converge on useful outcomes. The problem of how a global (cortical) optimization procedure can be implemented at a local (neuronal) level remains open.

To tackle the problem we turn to mechanism design: How to incentivize populations of rational agents to produce desirable outcomes? An inspiring successful application of mechanism design is prediction markets, which aggregate the behavior of self-interested traders into accurate predictions of diverse real-world events [8,18]. This has motivated research on payment schemes that encourage agents to trade in markets if the price distribution differs from their beliefs [14]. Of particular interest are proper scoring rules: payment schemes that incentivize rational agents to truthfully report their beliefs [17].

Our next step, §3, is therefore to analyze neuronal objective functions as payment schemes. This has implications in two directions. First, since the neuronal objective function decomposes as a sum over synapses, we model synapses as rational agents trading in a neuronal market, §3.1. Second, we model neurons as rational agents trading in a cortical prediction market, §4.1.3

Our main result, Theorem 5, establishes a striking connection between prediction markets and cortical learning: neuronal objective functions are proper scoring rules. The remainder of the paper applies two corollaries of Theorem 5 to show that well-functioning neuronal markets form a foundation for a well-functioning cortical market – thereby gluing together the two perspectives.

Corollary 6 shows that synaptic weights encode the utility expected after pre- and post- synaptic spikes. This partially answers the question posed earlier: “How does the brain encode information about the environment into its structure?”

More importantly, the corollary provides a foundation for cooperative learning. Consider the following basic schema to incentivize rational agents to collaborate:

(i) each agent estimates its usefulness to other agents,
(ii) incorporates the estimate into its reward function and
(iii) thus maximizes its usefulness to the collective.

To implement the schema, neurons must estimate their usefulness. Corollary 6 implies that synaptic weight $w_{ij}$ quantifies how useful spikes from $n_i$ are to $n_j$, when $n_j$ spikes. More generally, the sum of outgoing synaptic connections quantifies how useful a neuron’s outputs are to the rest of the system. We therefore define the usefulness of a neuron as, roughly, the sum of its downstream weights, §4.1.
In line with the schema we then show, Corollary 4 that incorporating feedback into reward functions causes neurons to (i) estimate their usefulness and (ii) maximize the estimate. This provides a new interpretation of a spike-based backpropagation scheme 20 that is closely related to error-backpropagation 21.

In short, well-functioning neuronal markets, with synapses faithfully reporting expected utilities, can be used to build well-functioning cortical markets.

Finally, experiments in 4 confirm our theoretical results.

Scope and related work.

A well-studied framework in neuroscience is based on the idea that neurons infer the probabilities of external events, which are encoded into probabilistic population codes, see e.g. 9. By contrast, we emphasize decisions over inferences. We are concerned with how neurons act, rather than what they infer. The two perspectives are related and it may turn out, as in prediction markets where prices can encode probabilities, that the population coding and mechanism design approaches lead to the same destination.

Note that our goal is to show methods from mechanism design can be fruitfully applied to fundamental questions in neuroscience. We do not advocate specifically for the scoring rules described below. These were derived from standard, but simple, neurophysiological models. With additional work it should be possible to extend our results to more realistic models.

2. A MINIMAL MODEL

At first glance, the models developed by neuroscientists are quite different from the rational agents studied in game theory. To build a bridge we utilize recent work discretizing a standard model from the neuroscience literature 5.

2.1 Discretized neurons

Consider a system of $N$ binary neurons $\{n_j\}_{j=1}^N$. Let $\mathcal{O} = \{0, 1\}^N$ denote the set of possible states. Each neuron is connected to a subset of the system. Suppose neuron $n_j$ has $K_j \ll N$ synapses. We model the restriction of the total state system to the subset received by neuron $n_j$ with a mask projecting from $\{0, 1\}^N$ to $\{0, 1\}^{K_j}$.

$$\varphi_j : \mathcal{O} \to \{0, 1\}^{K_j} : x = (x_1, \ldots, x_N) \mapsto (x_i)_{(i \in j \to j)}.$$ (1)

Neuron $n_j$ is equipped with a $K_j$-vector of synaptic weights, $w_j \in \mathcal{H}_j = \mathbb{R}^{K_j}$. Given input $x \in \mathcal{O}$, the neuron outputs a 0 or 1 according to

$$f_{w_j}(x) := \begin{cases} 1 & \text{if } \langle w_j, \varphi_j(x) \rangle - \vartheta > 0 \\ 0 & \text{else} \end{cases}$$ (2)

for some fixed $\vartheta$ constant across all neurons.

To simplify the exposition, we drop $\varphi_j$ from the notation and let $\mathcal{H} := \mathbb{R}^N$ denote the space of synaptic weights — where synapses that do not physically exist are implicitly clamped to zero. Thus, we treat entire system states as inputs to a neuron — when in fact the mask projects out most inputs.

Definition 1. Suppose we have utility function $\mu : \mathcal{O} \to \mathbb{R}$. Following 6, define reward function

$$R(x, w_j, \mu_j) = \mu_j(x) \cdot (\langle w_j, x \rangle - \vartheta) \cdot f_{w_j}(x)$$ (3)

Examples of utility functions are provided in 22 and 44.

Remark 1 (notation for spikes). Note that $f_{w_j}(x)$, $x_j$, $\mathbb{1}_{w_j}$, and $\mathbb{1}_j$ all denote the output of neuron $n_j$; emphasizing the function producing the output, that the output is also an input (one of many forming a vector) to other neurons, or the indicator-function aspect of the output respectively. We use $\mathbb{1}_{ij} := \mathbb{1}_i \cdot \mathbb{1}_j$ to indicate the co-spiking of neurons $n_i$ and $n_j$.

Ignoring costs for a moment, suppose neurons maximize $\mathbb{E}_{(x, \mu)} \left[ R(x, w_j, \mu) \right]$, where $P(x, \mu)$ is the joint distribution on spiking inputs and neuromodulators.

The reward function is continuously differentiable (in fact, linear) as a function of $w$ everywhere except at the kink $(w, x) = \vartheta$ where it is continuous but not differentiable. We can therefore perform gradient ascent to obtain synaptic updates

$$\Delta w_{ij} \propto \mu_j(x) \cdot x \cdot f_{w_j}(x) = \mu_j(x) \cdot \mathbb{1}_{ij}. \quad (4)$$

In short, if $n_j$ receives input $\mathbb{1}_i$ and subsequently spikes $\mathbb{1}_j$, then synapse $i \to j$ is modified proportionally to $\mu_j(x)$. The main theorem in 5 derives the above equations by discretizing standard models of neuronal dynamics and learning:

Theorem 1 (discretized neurons, 5). The fast time constant limit of Gerstner’s Spike Response Model 13 is 6. Taking the fast time constant limit of STDP 23 yields 4 with $\mu_j(x) = 1$. Finally, STDP is gradient ascent on a reward function whose limit is 4.

Spike-timing dependent plasticity is prone to overpotentiation 23, leading to epileptic seizures. In the neuroscience literature, weights are typically controlled with a deotentiation bias. We take an alternative approach, by introducing a regularizer $A_\mu(w)$, which quantifies the resource costs incurred by high synaptic weights 30 16.

The optimal weights are then computed according to

$$w_j^* := \arg \max_{w \in \mathcal{H}} \mathbb{E}_P \left[ S_\mu(x; w) \right] \quad (5)$$

$$:= \arg \max_{w \in \mathcal{H}} \mathbb{E}_P \left[ R(x, w, \mu) - A_\mu(w) \right] \quad (6)$$

where scoring rule $S_\mu(x, w)$ balances rewards $R(x, w, \mu)$ against costs $A_\mu(w)$.

We consider two standard regularizers taken from machine learning 22 and a third, more biologically plausible, taken from 5

$$\begin{align*}
A_2(w_j) &= \frac{1}{2\eta} \| w_j \|^2 \\
A_1(w_j) &= \frac{1}{\eta} \sum_{i} w_{ij} \log w_{ij} \\
A_1(w_j) &= \frac{1}{\eta} \| w_j \|_1, \text{ where } 0 \leq w_{ij} \leq 1 \text{ for all } i. \quad \ell_2 \quad \ell_1 \quad \ell_1
\end{align*}$$

Clearly $\ell_2$ is not a norm — we find the notation convenient.

Computing gradient ascent yields online updates

$$\Delta w_{ij} \propto \mu_j(x) \cdot \mathbb{1}_{ij} - \frac{1}{\eta} \begin{cases} w_{ij} & \ell_2 \\
\log w_{ij} + 1 & \ell_H
\end{cases} \quad (7)$$

Remark 2 (regularizers). Each regularizer has points in its favor. The $\ell_1$ regularizer provides a simple interpretation of the saturated synaptic weights observed in some neurophysiological models 12. The $\ell_2$ regularizer allows negative synaptic weights, corresponding to inhibitory synapses.
Finally, \( \ell_{ij} \) results in weights that can be interpreted as a probability distribution and is closely related to Hanson's logarithmic market scoring rule \([12]\).

2.2 Utility functions

Three biologically inspired utility functions are:

**Example U1.** (Feedforward, frequency). Utility function \( \mu(x) = 1 \) encourages neurons to spike for inputs that are frequent and contain many spikes.

**Example U2.** (Feedforward, invariance). A more interesting utility function takes inputs over consecutive time steps \( x = \langle x^{(t-1)}, x^{(t)} \rangle \) as input and sets \( \mu(x) = f_w(x^{(t-1)}) \). This encourages neurons to learn stable patterns containing many spikes, i.e., those that cause it to spike twice consecutively. The utility function can be extended across multiple time steps, possibly with a temporal discount factor.

**Example U3.** (Neuromodulators). Neuronal modulatory systems signaling global rewards can be modeled via \( P(\nu|x) \) where \( \nu \) is a real-valued random variable: positive outcomes are reinforced and conversely. The utility is then \( \mu(x) := \mathbb{E}_{\nu \sim P(\nu|x)}[\nu|x] \), where the expectation is with respect to the distribution on neuromodulators.

A fourth utility function is discussed in \([11]\).

3. NEURONAL PREDICTION MARKETS

Scoring rules are schemes for paying agents based on their reports. Proper scoring rules, which incentivize agents to report truthfully, have proven useful in a wide range of settings including weather forecasts \([10]\), prediction markets \([14, 17]\) and crowdsourcing and learning mechanisms \([2,3]\).

Our main result, Theorem 5, is that the scoring rules \( S_\nu \) in \([6]\) are proper for three regularizers. The upshot is that a neuron’s synaptic weights faithfully encode expectations about rewards after pre- and post-synaptic spiking activity. The form of the encoding depends on the regularizer.

3.1 Synapses as rational agents

This subsection argues that synapses are analogous to traders, operating within a neuronal market, that attempt to maximize their payout relative to their expenditures.

Prediction market traders buy and sell contingent securities, which pays out $1 if an outcome belongs to a particular set, and $0 otherwise \([4]\). For example an Arrow-Debreu security could pay $1 if an outcome belongs to a particular set, and $0 otherwise. For example an Arrow-Debreu security could pay $1 if an outcome belongs to a particular set, and $0 otherwise. Given functions \( \rho: \mathcal{O} \rightarrow \mathcal{H} \) and \( F: \mathcal{H} \rightarrow \mathbb{R} \), define

\[
S_{\rho} : \mathcal{O} \times \mathcal{H} \rightarrow \mathbb{R} : (x; w) \mapsto -D_{\mathcal{F}}(\rho(x), w) - F(\rho(x))
\]

where \( D_{\mathcal{F}}(x, y) := : F(x) - F(y) - \langle \nabla_{\mathcal{F}}(y), x - y \rangle \) is the Bregman divergence. It is shown in \([3]\) that:

**Proposition 2** (Linear proper scoring rules). If \( F \) is convex then \( S_{\rho} \) is a proper scoring rule for linear property \( \Gamma: P \mapsto \mathbb{E}_{P}[\rho(x)] \).

3.2 Proper scoring rules

The remainder of this section uses properness to precisely quantify how synaptic weights relate to utility expectations.

**Definition 2.** Let \( \mathcal{P}_O \) be a set of probability distributions on states \( \mathcal{O} \) and define a property \( as a function \( \Gamma : \mathcal{P}_O \rightarrow \mathcal{H} \). Scoring rule \( S : \mathcal{O} \times \mathcal{H} \rightarrow \mathbb{R} \) is proper \( \Gamma \) if for all \( P \in \mathcal{P}_O \)

\[
\Gamma(P) \in \arg\max_{w \in \mathcal{R}(\Gamma)} \mathbb{E}_P[S(x; w)].
\]

Properness is the common-sensical requirement that the true value, \( w = \Gamma(P) \), is a score maximizer, \( w \in \arg\max \mathbb{E}_P[S] \). In short: “you get what you think you are paying for”.

Proper scoring rules can be constructed as follows \([3]\).

Given functions \( \rho: \mathcal{O} \rightarrow \mathcal{H} \) and \( F: \mathcal{H} \rightarrow \mathbb{R} \), define

\[
S_{\rho} : \mathcal{O} \times \mathcal{H} \rightarrow \mathbb{R} : (x; w) \mapsto -D_{\mathcal{F}}(\rho(x), w) - F(\rho(x))
\]

where \( D_{\mathcal{F}}(x, y) := : F(x) - F(y) - \langle \nabla_{\mathcal{F}}(y), x - y \rangle \) is the Bregman divergence. It is shown in \([3]\) that:

**Proposition 2** (Linear proper scoring rules). If \( F \) is convex then \( S_{\rho} \) is a proper scoring rule for linear property \( \Gamma: P \mapsto \mathbb{E}_{P}[\rho(x)] \).

3.3 Proper scoring for discretized neurons

This section adapts Proposition 2 to discretized neurons. As a warmup, we show that the selectivity term from \([4]\) yields proper scoring rules.

**Lemma 3.** Let \( T_{\rho}(x; w_j) := \mu_j(x) \cdot \langle (w_j, x) - \theta \rangle - A_*(w_j) \) be scoring rules. These are proper for \( \Gamma^T : \mathcal{P}_O \rightarrow \mathcal{H} = \mathbb{R}^N \),

\[
\Gamma^T : P \mapsto G_*\left( \mathbb{E}_P[\mu_j(x) \cdot x] \right) \quad \text{for} \quad G_{\mathbb{H}}(v) = e^{-v-1}, \quad G_1(v) = \mathbb{1}_{v \geq 1},
\]

where \( \mathbb{1}_{v \geq 1} \) an N-vector of indicator functions returning 1 when \( v > 1 \) and 0 otherwise.
Proof. We drop $\vartheta$ since it is independent of $w_j$. Define hypothesis space $\mathcal{H} = \mathbb{R}^N$ and map

$$\rho_0 : \mathcal{O} \rightarrow \mathcal{H} : x \mapsto \mu(x) \cdot x.$$  

We consider the three cases in turn.

Observe that convex function $F_2(x) = \frac{1}{2\eta} \| x \|_2^2$ yields a scoring rule $S_2(x, w_j) = \langle \mu(x) \cdot x, w_j \rangle - \frac{1}{2\eta} \| w_j \|_2^2$, which implies $T_2$ is proper by Proposition 2.

For $T_H$, restrict $\mathcal{H}$ to the subset of $\mathbb{R}^N$ where $\sum_i \exp(\frac{1}{\eta} w_i x_i) = 1$ and define $\psi : \mathcal{H} \rightarrow \mathbb{R}^N$ taking $w_j \mapsto \exp(\frac{1}{\eta} w_j x_j)$. Convex function $F_H(x) = \eta \log(\sum_i \exp(\frac{1}{\eta} x_i))$ yields

$$S_H(x, w_j) = F_H(w_j) + \langle \psi(w_j), \mu_j(x) \cdot x - w_j \rangle$$

$$= \langle \psi(w_j), \mu_j(x) \cdot x \rangle - \frac{1}{\eta} \langle \psi(w_j), \log(\psi(w_j)) \rangle$$

since $F_H(w_j) = 0$. By Proposition 2 it follows that $T_H$ is proper for linear property $E[\eta \cdot v]$. The result follows for $\Gamma_H$ since $e^{-1}$ is monotonically. We use the nonlinear “exp” representation since it directly corresponds to synaptic weights which will be useful in Theorem 3.

Proposition 2 does not apply to $T_1$, so we derive properness by other means. Computing gradients obtains

$$\Delta w_j \propto E_P[\mu_j(x) \cdot x] - \frac{1}{\eta}$$

which has a stationary point when all synaptic weights are equal to the scalar $\eta$. The stationary point is unstable - a local minima rather than maxima. Synapses with $E_P[\eta \mu_j(x) \cdot 1] > 1$ are forced to boundary condition $w_j = 1$; others are forced to 0 (for simplicity we assume no expectation is precisely 1).

The range of $\Gamma^T_1$ is the set of $N$-vectors of 0s and 1s. Any $w \in \text{range}(\Gamma^T_1)$ differing from $\Gamma^T_1(P)$ has non-zero gradient and hence a lower score, implying $\Gamma^T_1$ is proper.

The selectivity term in [3] introduces a complication into the scoring rule: potentiating a synaptic weight may cause a neuron to stumble over a change in its utility function that is hidden by the selectivity term. Although the reward function is continuous in $w$ its derivative is not: there is a kink. We bound the jump after crossing a kink via

**Assumption 1 (No nasty surprises).** If $\Delta_{ij} > 0$ then $\exists \epsilon > 0$ such that $E_P[\mu_j(x) \cdot 1_{\Delta_{ij}}] > \epsilon$, where $1_{\Delta_{ij}} := 1_{w_j + \epsilon \Delta_{ij}} - 1_{w_j}$.

Assumption 1 implies that sufficiently small synaptic updates, Eq. [7], always increase a neuron’s score.

**Lemma 4 (Smooth ascent).** Under Assumption 1 if $\Delta_{ij} > 0$ then $\exists \epsilon > 0$ such that

$$E_P[S_1(x, w_j + \epsilon \cdot \Delta_{ij})] > E_P[S_1(x, w_j)]$$

and similarly for $\Delta_{ij} < 0$.

**Proof.** Straightforward computation.

Informally, if high utility follows $n_i$ and $n_j$ cospiking, then Assumption 1 says that the utility of new inputs, causing $n_j$ to spike when synapse $i$ increases by $\Delta_{ij}$, is not too negative. If the assumption fails then the neuron will continuously potentiate and depotentiate synapse $i$ as the gradient jumps from positive to negative. This is analogous to the behavior of a perceptron when confronted with classes that are not linearly separable.

Nasty surprises can be avoided in at least two ways. First, by designing the utility function so that it behaves well with respect to the distribution the neuron encounters. Second, by allowing neuron $n_i$ to modify its regularization parameter $\eta_j$. Going further, one could introduce additional degrees of freedom by associating an $\eta_j$ with each synapse (note the regularizers are sums over synapses) that is tweaked when a neuron detects that one of its synapses jumps back and forth. We do not pursue these ideas here.

Before proving our main result, we introduce some notation. Given $w$, let $1_w := f_w^{-1}(1) \subset O$ and let $2^O$ denote the powerset of $O$. Enlarge the hypothesis space to $\mathcal{H}' := 2^O \times \mathcal{H}$ with embedding $\psi : \mathcal{H} \rightarrow \mathcal{H}' : w \mapsto (1_w, w)$. Let $w^* = \text{argmax}_{w \in C} E_P[S_1(x)]$.

**Theorem 5 (Neuronal scoring rules are proper).** Under Assumption 1 scoring rules $S_*$ are proper for property $\Gamma_* : P \rightarrow \mathcal{H}' = 2^O \times \mathcal{H}$, $\Gamma_* : P \mapsto (1_{w^*}, G_*(E_P[\mu(x) \cdot x \cdot 1_{w^*}]))$, (10)

where $G_*(x) \in \{\eta, e^\eta \cdot v^{-1}, 1_{\eta \cdot v > 1}\}$ depending on the choice of regularizer.

**Proof.** Property $P \mapsto 1_{w^*}$ is proper by construction; we therefore focus on the synaptic term $G_*(\cdot)$ in Eq. 10.

Computing gradients for $S_2$ and $S_H$ yields stationary points

$$w^* = E_P[\eta \cdot \mu(x) \cdot x \cdot 1_{w^*}]$$

$$w^* \propto \exp \left( E_P[\eta \cdot \mu(x) \cdot x \cdot 1_{w^*}] - 1 \right)$$

respectively which are stable maxima under Assumption 1 by Lemma 4. As argued before, a weight vector $w$ that does not have zero-gradient cannot be a maxima, and the argument follows from Lemma 3.

Similar reasoning applies to $S_1$.

**Remark 3 (Indirect elicitation).** Eliciting properties from distributions was studied in [17], which draw a distinction between elicitable and directly elicitable properties. For example, the variance can only be elicited by a scoring rule if the mean is elicited as well. Similarly, $G_*(E_P[\mu(x) \cdot 1_{w^*}])$ cannot be elicited directly, but only in conjunction with $1_{w^*}$.

Neurons only modify their synapses to incorporate rewards when spiking. Eq. [4]. This encourages specialization, but also implies that individual neurons may never discover that spiking for certain inputs results in very high utility. More formally, the kink makes $S_*$ non-convex, so gradient ascent is not guaranteed to find the global optimum.

Nevertheless, the relationship between synaptic weights and expected utilities in Theorem 3 still holds:

**Corollary 6 (Synaptic code).** Let $\bar{w}$ be the (in general local) maximum of $S_*$ obtained by gradient ascent with Eq. [7]. If Assumption 1 holds then $\bar{w}$ satisfies

$$\bar{w} = G_*(E_P[\mu(x) \cdot x \cdot 1_{w}]).$$ (11)
Note that Eq. (11) is not in closed form since \( \hat{w} \) appears on both the left- and right-hand sides.

Proof. Since local maxima are stationary points, the proof follows the same argument as Theorem 3. \( \square \)

A discretized neuron \( n_j \) thus faithfully encodes two properties of its input distribution. First, its spikes encode a set of inputs for which spiking is locally optimal. Second, its synaptic weights encode the expected utility per synapse when \( n_i \) and \( n_j \) co-spike.

Remark 4 (neural code). Corollary 2 provides an interesting interpretation of the meaning of spikes. A neuron spikes if the dot product \( (w_j, x) \) is above threshold \( \vartheta \). That is, neuron \( n_j \)'s spike means that the current system state \( x \) is significant (above threshold) when evaluated against the utility expectations \( w_j \) that were previously encoded into \( n_j \)'s structure.

Similarly to how stock price movements encode information about which sectors of an economy are expected to yield high profits in the near future; spikes and synaptic weights encode expectations about future rewards.

4. CORTICAL PREDICTION MARKETS

This section investigates how neurons can estimate their usefulness to downstream neurons, and so allocate their resources such that the benefit to other neurons is maximized. In short, we introduce a utility function that incentivizes neurons to optimize their usefulness to other neurons.

4.1 Backpropagation: errors or incentives?

To provide context, we recall related work on incorporating spikes into a reward signal. Neuromodulators provide a primary reward system. However, neurons whose actions do not directly result in pleasure or pain may require more indirect incentives. In machine learning, multilayer networks are often trained by backpropagating errors [21]. However, backpropagation (BP) is biologically implausible — it requires pathways for backpropagating errors which have not been observed in cortex [20].

As an alternative, [20] proposed attention-gated reinforcement learning (AGREL), which uses feedback spikes as attention signals to modulate learning. AGREL abstracts two features of feedback (NMDA) connections in cortex: (i) they prolong, but do not initiate, spiking activity and (ii) they co-spike with downstream populations, \( x^f \) and \( x^b \) respectively, and define \( H^f \) and \( H^b \) by clamping weights in the respective populations to zero using Eq. (1). Define the feedback utility as \( \mu^b_j(x) := (w^b_j, x) \) for \( w^b_j \in H^b \).

A neuron with feedback utility maximizes

\[
E_F \left[ \langle (w^b_j, x) - \vartheta \rangle 1_j - A_*(w_j) \right] \quad (13)
\]

and so aligns its feedforward \( (w^f_j, x) \) and feedback \( (w^b_j, x) \) current whenever the neuron itself spikes.

Computing gradient ascent on scoring rule \( S^b_j(w; x) = (w^b_j, x) \cdot ((w^f_j, x) - \vartheta) \cdot 1_j - A_*(w) \) obtains

\[
\Delta w^f_{ij} \propto (w^f_j, x) \cdot 1_{ij} - \partial A_*(w), \quad (14)
\]

which differs from AGREL [12] by using \( \partial A_* \) as regularizer instead of \( (1 - x_j) \) and extending feedback from a single neuron, \( w^b_j x_k \), to many neurons, \( (w^b_j, x) \). We also drop the global reward signal \( f(\delta) \) since we are interested in the pure backpropagation case; it can easily be reinstated.

Note that the utility function \( \mu^b_j \) is itself plastic. Neuron \( n_j \) not only modifies feedforward weights to maximize its score, it also modifies feedback weights to increase the maximum achievable score:

\[
\Delta w^b_{ij} \propto ((w^f_j, x) - \vartheta) \cdot 1_{ij} - \partial A_*(w). \quad (15)
\]

4.2 Estimating usefulness with feedback

As suggested in the introduction, one way to encourage collaboration is for each neuron to estimate its usefulness to the rest of the system and optimize that estimate. By Corollary 3, a faithful measure of the usefulness of \( n_j \)'s output to the rest of cortex is the sum of active downstream synaptic weights:

\[
V_j(x) := \sum_{(k \land j = k)} w_{jk} 1_{jk} = (w_{j*}, x) 1_j. \quad (16)
\]

Intuitively, \( V_j(x) \) is the total utility that spiking downstream neurons expect after \( n_j \) spikes.

Neurons cannot compute their usefulness directly, since the utilities of downstream neurons are private. They must therefore make do with publicly available data: spikes by other neurons. We therefore propose that neurons use feedback, which they can actually compute, as a proxy for usefulness, which would be ideal.

As a consequence of Corollary 6, we quantify how closely feedback-utility approximates usefulness:

\[
\sum_k 1_{jk} G_* \left( \mathbb{E} [\mu_j(x) 1_{jk}] \right) - G_* \left( \mathbb{E} [(w^f_j, x) 1_{jk}] \right). \]

Example U4. (Feedback). Identify disjoint upstream and downstream populations, \( x^f \) and \( x^b \), respectively, and define \( H^f \) and \( H^b \) by clamping weights in the respective populations to zero using Eq. (1). Define the feedback utility as \( \mu^b_j(x) := (w^b_j, x) \) for \( w^b_j \in H^b \).

A utility function that incentivizes neurons to optimize their usefulness to other neurons.
Thus, the quality of \( (w^f_k, x) \) as an estimate of \( V_j(x) \) depends on how closely \( n_j \)'s feedforward inputs \( (w^f_j, x) \) approximate the sum of the downstream utilities \( \mu_j(x) \).

**Proof.** The usefulness and utility of \( n_j \) are
\[
V_j(x) = (w_j, x) \mathbb{1}_j \quad \text{and} \quad \mu_j^f(x) = (w^f_j, x)
\]
respectively.

The utility \( \mu^f_j \) is multiplied by \( \mathbb{1}_j \) when it is used in scoring rules, so the difference comes down to the weights. Corollary 6 implies the optimal feedforward weights are
\[
w^f_j = G_S \left( E_P [\mu_k(x) \cdot \mathbb{1}_{jk}] \right)
\]
so that the usefulness of \( n_j \) is
\[
V_j(x) = \sum_{k \neq j} \mathbb{1}_j \cdot G_S \left( E_P [\mu_k(x) \mathbb{1}_{jk}] \right).
\]

The utility function is a proxy for the cost term, Theorem 1:
\[
\text{V} \left( \mathbb{1}_j \right) = G_S \left( E_P [\mu_k(x) \cdot \mathbb{1}_{jk}] \right)
\]

and we are done.

Experiments in [5] demonstrate that \( w^f \) is a good proxy for \( w_{jk} \) in some interesting cases.

A detailed analysis of the relationship between approximations \( w_{jk} \approx w^f_{jk} \), distribution \( P(x) \), and utility functions \( \mu_k(x) \) is beyond the scope of this paper.

### 4.3 Neurons as rational agents

Section 3.1 suggested neurons are analogous to markets in which synapses trade. This section presents a second analogy, where cortex forms a market in which neurons trade.

Recall that neurons are rational agents that optimize their expected reward balanced against a cost term, Theorem 1:
\[
w^* \left( \mathbb{1}_j \right) = \arg\max_{w_{jk} \in H} E_P \left[ \left( (w_j, x) - \theta \right) \cdot \mu_j(x) \mathbb{1}_j - A_\ast(w_j) \right].
\]

The key idea is that each neuron should optimize its usefulness to the rest of the brain. Building on Corollary 6, usefulness is defined as \( V_j(x) = \sum_{k \neq j} w_{jk} \mathbb{1}_{jk} \). That is, the quantity of \( \mathbb{1}_j \) used by downstream neurons in their internal markets. Unfortunately, \( n_j \) does not have access to this number. Similarly to how musicians are paid for actual sales rather than downloads of their music, neurons need to record when their outputs are used. They therefore use feedback to compute \( \left( w^f_{jk}, x \right) \mathbb{1}_j \), which acts as a proxy\(^2\) for \( \left( w^f_{jk}, x \right) \mathbb{1}_j \).

Intuitively, \( n_j \) simultaneously sets its feedback connections on the downstream traders that most frequently purchase its spikes, and sets its feedforward connections on the upstream traders that sell the most useful spikes.

The result is a mesh of intertwining neuronal chains – optimized for usefulness at every link by the invisible hand of the cortical market – that connects sensory inputs to motor actions.

\(^2\)Recorded usage could over- or under- estimate true usage. Section 5 shows that it is a good guide in practice.

### 5. EXPERIMENTS

We investigate the empirical performance of discretized neurons. The experiments are designed to show that: (i) the ideas above can be implemented with minimal modifications; (ii) synaptic weights encode environmental statistics and rewards; (iii) feedback improves performance; and (iv) feedback reliably estimates a neuron’s usefulness.

We have therefore constructed networks, inspired by [10], that learn tasks designed so that the embedding of expected utilities into synaptic weights is easy to visualize.

Our goal is not to compete with the state of the art. Rather, our aim is to introduce mechanism design techniques into the analysis and construction of networks. A pressing open question is whether more sophisticated networks, such as those developed by the deep learning community, can be understood or improved via mechanism design.

**Network architectures.**

The tracker network, Fig. 1 left, has a sensory grid \( S \) of \( 20 \times 20 \) neurons, intermediate layers \( V \) and \( D \) with 100 neurons each, motor layer \( M \), and 100 randomly connected inhibitory neurons \( INH \). Signals from \( S \) to \( D \) are delayed so that \( V \) and \( D \) receive different temporal snapshots of \( S \). Synapses are plastic except those to or from inhibitory neurons. \( M \) is divided into 8 areas of 10 neurons each. Actuators engage when they receive more than 10 spikes. The network is initialized randomly.

The tracker network tracks targets traveling along an edge of the visual field. Motor areas are rewarded \( (\mu_j = +1) \) or punished \( (\mu_j = -1) \) according to whether or not the action correctly anticipates where the target is headed and from which direction (4 \( \times \) 2 possibilities). Note the motor layer receives neuromodulatory signals whereas the intermediary layers do not and learn from feedback.

The foveator network, Fig. 1 right, drops \( D \) and has fewer inhibitory neurons.

The foveation task is to move the fovea (center of the retina) onto an object appearing on the edge of the visual field. Each motor area controls an actuator that moves the fovea in a compass direction (N, NW, W, etc). After a movement, the corresponding area is rewarded \( (\mu_j = 1) \) if the object is closer to the center and punished \( (\mu_j = -1) \) otherwise.

We tweak the discretized neuron in [1] to make the dynamics closer to continuous time models of cortical neurons.

First, we introduce a voltage term \( V \), which provides neurons with a steadily decaying “memory” of previously received spikes. Neurons spike when \( V > \vartheta \), after which \( V \leftarrow 0 \). When the neuron does not spike, \( V \) is updated.
according to
\[ V \leftarrow V + (w, x) - \delta. \]

Neurons maintain an exponentially decaying trace reflecting recent output spikes:
\[ \text{trace}_j \leftarrow 0.95 \cdot (\text{trace}_j + 0.4 \cdot \mathbb{1}_j) \]

Neurons in subsystems \( V \) and \( D \) update their feedforward synapses according to
\[ \Delta w_{ij} \propto \langle w_{fb}, x \rangle \cdot \mathbb{1}_i \cdot \text{trace}_j \]
and similarly for feedback. Thus, \( \text{trace}_j \) is substituted for \( \mathbb{1}_j \) to temporally smooth out learning. Neurons in subsystem \( M \) update their synapses according to
\[ \Delta w_{ij} \propto \mu_j(x) \cdot \sum_{t'=t-m}^{t} \mathbb{1}_{ij} \]
where the sum is over tics since the last neuromodulatory signal, similar to the trace implemented in \( V \) and \( D \).

Finally, we tweak the regularization. Instead of continuously regularizing by \( A_1(w) \), we regularize at discrete intervals, analogous to a hypothesized role of sleep [6]. Regularization consists of setting the \( K \) strongest synapses to 1, and pruning the rest (i.e. setting their weights to 0). The number \( K \) is fixed within each layer, but varies across layers.

Visualizing synaptic strengths.

By Corollary 7, the quality of a neuron’s usefulness-estimate can be computed and visualized by comparing average feedforward and feedback weights.

Weights are visualized in Fig 2 and 3 as follows. For each area in \( M \), we average over all feedforward paths \( S \to V \to M \) and feedback paths \( S \to V \leftarrow M \) respectively, and similarly for \( D \). To save space, 3 out of 8 areas in \( M \) are plotted. Plots are averaged over 20 runs. Blue denotes low values; red denotes high.

**Results.**

(i) The tasks are easy and the networks rapidly (within a few thousand tics) achieve 98% and 95% accuracy. The tracker outperforms the foveator, possibly because the foveator modifies its environment by actively moving the center of the retina, whereas the tracker does not.

The delay line is essential to tracker performance: if the delay is set to zero then the network performs little better than chance, and the structure of the environment is not learned at all.

<table>
<thead>
<tr>
<th>Tracker</th>
<th>% correct</th>
<th># correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>only M plastic</td>
<td>95</td>
<td>243</td>
</tr>
<tr>
<td>all plastic</td>
<td>98</td>
<td>672</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Foveator</th>
<th>% correct</th>
<th># correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>only M plastic</td>
<td>93</td>
<td>80</td>
</tr>
<tr>
<td>all plastic</td>
<td>95</td>
<td>207</td>
</tr>
</tbody>
</table>

(ii) The middle rows of Fig 2 and 3 show how rewards and environmental statistics are incorporated into the networks’ feedforward structure.

For the tracker network, the \( V \)-area synapses learn trajectories; whereas the \( D \)-area learns the starting points of trajectories. The combination of instantaneous lines in \( V \) (which learn directions) and delay lines in \( D \) (which learn starting points) thus allows the network to implicitly compute derivatives and thereby determine directions of travel.

For the foveator, it is easy to read off the correspondence between the NE, N, and NW movements of the actuators and the locations of objects driving the movements.

(iii) Shutting off feedback plasticity (top rows of Fig 2 and 3) slightly worsens performance, from 98% to 95% for the tracker and from 95% to 93% for the foveator. However, it dramatically worsens the “reaction times” of the networks, quantified as the number of times the actuators correctly engage per 1000 tics.

Indeed, looking at synaptic weights without feedback plasticity, top rows of the figures, we find that the structure of the rewards and environment is barely visible.

(iv) Finally, when feedback plasticity is turned on, average synaptic weights over feedforward paths \( T \to V/D \to M \) (middle rows) and feedback paths \( T \to V/D \leftarrow M \) (bottom rows) are almost identical, demonstrating that neurons in \( V \) and \( D \) accurately estimate their usefulness to downstream \( M \) neurons using feedback.
6. CONCLUSION

This paper applied tools from mechanism design to investigate a simple model of cortical neurons. The main result is that, under a technical assumption, neurons faithfully encode expected utilities into their synaptic weights. If the result can be extended to more realistic models, then it will provide a powerful new approach to understanding the relationship between cortical structure and function.

There is good reason to be optimistic: extending the analysis to continuous time models requires exponential discount factors analogous to interest rates – which are well-understood in mechanism design.

An important corollary of the analysis is a novel interpretation of the role of spiking feedback in cortex: neurons can use feedback spikes to estimate their usefulness to the rest of cortex, and then learn to maximize that estimate.

We have used the the simplest possible scoring rules, derived from standard models of neurons, to provide a proof of principle. It will be interesting to explore more realistic models taken from the neuroscience literature, and also more powerful models such as those developed for deep neural networks.

Finally, although the flow of ideas in this paper is one-sided — from mechanism design to neuronal models — we expect that future work will be more symmetric. The cortex aggregates information far more effectively than the auctions and online markets studied in game theory. This suggests there are powerful design principles waiting to be uncovered.

Acknowledgements. I am grateful to Hastagiri Vanchinathan for encouraging me to look into mechanism design.

7. REFERENCES


