Deep Learning Games

Dale Schuurmans[*]
Google
daes@ualberta.ca

Martin Zinkevich
Google
martinz@google.com

Abstract

We investigate a reduction of supervised learning to game playing that reveals new connections and learning methods. For convex one-layer problems, we demonstrate an equivalence between global minimizers of the training problem and Nash equilibria in a simple game. We then show how the game can be extended to general acyclic neural networks with differentiable convex gates, establishing a bijection between the Nash equilibria and critical (or KKT) points of the deep learning problem. Based on these connections we investigate alternative learning methods, and find that regret matching can achieve competitive training performance while producing sparser models than current deep learning strategies.

1 Introduction

In this paper, we investigate a new approach to reducing supervised learning to game playing. Unlike well known reductions [9, 32, 33], we avoid duality as a necessary component in the reduction, which allows a more flexible perspective that can be extended to deep models. An interesting finding is that the no-regret strategies used to solve large-scale games [39] provide effective stochastic training methods for supervised learning problems. In particular, regret matching [13], a step-size free algorithm, appears capable of efficient stochastic optimization performance in practice.

A central contribution of this paper is to demonstrate how supervised learning of a directed acyclic neural network with differentiable convex gates can be expressed as a simultaneous move game with simple player actions and utilities. For variations of the learning problem (i.e. whether regularization is considered) we establish connections between the critical points (or KKT points) and Nash equilibria in the corresponding game. As expected, deep learning games are not simple, since even approximately training deep models is hard in the worst case [15]. Nevertheless, the reduction reveals new possibilities for training deep models that have not been previously considered. In particular, we discover that regret matching with simple initialization can offer competitive training performance compared to state-of-the-art deep learning heuristics while providing sparser solutions.

Recently, we have become aware of unpublished work [2] that also proposes a reduction of supervised deep learning to game playing. Although the reduction presented in this paper was developed independently, we acknowledge that others have also begun to consider the connection between deep learning and game theory. We compare these two specific reductions in Appendix J, and outline the distinct advantages of the approach developed in this paper.

2 One-Layer Learning Games

We start by considering the simpler one-layer case, which allows us to introduce the key concepts that will then be extended to deep models. Consider the standard supervised learning problem where one is given a set of paired data $\{(x_t, y_t)\}_{t=1}^T$, such that $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$, and wishes to learn a

[*] Work performed at Google Brain while on a sabbatical leave from the University of Alberta.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

predictor $h:\mathcal{X}\to\mathcal{Y}$. For simplicity, we assume $\mathcal{X} = \mathbb{R}^m$ and $\mathcal{Y} = \mathbb{R}^n$. A standard generalized linear model can be expressed as $h(x) = \phi(\theta x)$ for some output transfer function $\phi : \mathbb{R}^n \to \mathbb{R}^n$ and matrix $\theta \in \mathbb{R}^{n\times m}$ denoting the trainable parameters of the model. Despite the presence of the transfer function ϕ, such models are typically trained by minimizing an objective that is convex in $z = \theta x$.

OLP (One-layer Learning Problem) Given a loss function $\ell : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ that is convex in the first argument, let $\ell_t(z) = \ell(z, y_t)$ and $L_t(\theta) = \ell_t(\theta x_t)$. The training problem is to minimize $L(\theta) = T^{-1}\sum_{t=1}^{T} L_t(\theta)$ with respect to the parameters θ.

We first identify a simple game whose Nash equilibria correspond to global minima of the one-layer learning problem. This basic relationship establishes a connection between supervised learning and game playing that we will exploit below. Although this reduction is not a significant contribution by itself, the one-layer case allows us to introduce some key concepts that we will deploy later when considering deep neural networks. A one-shot **simultaneous move game** is defined by specifying: a set of players, a set of actions for each player, and a set of utility functions that specify the value to each player given a joint action selection [40, Page 9] (also see Appendix E). Corresponding to the OLP specified above, we propose the following game.

OLG (One-layer Learning Game) There are two players, a protagonist p and an antagonist a. The protagonist chooses a parameter matrix $\theta \in \mathbb{R}^{m\times n}$. The antagonist chooses a set of T vectors and scalars $\{a_t, b_t\}_{t=1}^{T}$, $a_t \in \mathbb{R}^n$, $b_t \in \mathbb{R}$, such that $a_t^\top z + b_t \le \ell_t(z)$ for all $z \in \mathbb{R}^n$; that is, the antagonist chooses an *affine minorant* of the local loss for each training example. Both players make their action choice without knowledge of the other player's choice. Given a joint action selection $(\theta, \{a_t, b_t\})$ we define the utility of the antagonist as $U^a = T^{-1}\sum_{t=1}^{T} a_t^\top \theta x_t + b_t$, and the utility of the protagonist as $U^p = -U^a$. This is a two-person zero-sum game with continuous actions.

A **Nash equilibrium** is defined by a joint assignment of actions such that no player has any incentive to deviate. That is, if $\sigma^p = \theta$ denotes the action choice for the protagonist and $\sigma^a = \{a_t, b_t\}$ the choice for the antagonist, then the joint action $\sigma = (\sigma^p, \sigma^a)$ is a Nash equilibrium if $U^p(\tilde{\sigma}^p, \sigma^a) \le U^p(\sigma^p, \sigma^a)$ for all $\tilde{\sigma}^p$, and $U^a(\sigma^p, \tilde{\sigma}^a) \le U^a(\sigma^p, \sigma^a)$ for all $\tilde{\sigma}^a$.

Using this characterization one can then determine a bijection between the Nash equilibria of the OLG and the global minimizers of the OLP.

Theorem 1 *(1) If $(\theta^*, \{a_t, b_t\})$ is a Nash equilibrium of the OLG, then θ^* must be a global minimum of the OLP. (2) If θ^* is a global minimizer of the OLP, then there exists an antagonist strategy $\{a_t, b_t\}$ such that $(\theta^*, \{a_t, b_t\})$ is a Nash equilibrium of the OLG.* **(All proofs are given in the appendix.)**

Thus far, we have ignored the fact that it is important to control model complexity to improve generalization, not merely minimize the loss. Although model complexity is normally controlled by regularizing θ, we will find it more convenient to equivalently introduce a constraint $\theta \in \Theta$ for some convex set Θ (which we assume satisfies an appropriate constraint qualification; see Appendix C). The learning problem and corresponding game can then be modified accordingly while still preserving the bijection between their solution concepts.

OCP (One-layer Constrained Learning Problem) Add optimization constraint $\theta \in \Theta$ to the OLP.
OCG (One-layer Constrained Learning Game) Add protagonist action constraint $\theta \in \Theta$ to OLG.

Theorem 2 *(1) If $(\theta^*, \{a_t, b_t\})$ is a Nash equilibrium of the OCG, then θ^* must be a constrained global minimum of the OCP. (2) If θ^* is a constrained global minimizer of the OCP, then there exists an antagonist strategy $\{a_t, b_t\}$ such that $(\theta^*, \{a_t, b_t\})$ is a Nash equilibrium of the OCG.*

2.1 Learning Algorithms

The tight connection between convex learning and two-person zero-sum games raises the question of whether techniques for finding Nash equilibria might offer alternative training approaches. Surprisingly, the answer appears to be yes.

There has been substantial progress in on-line algorithms for finding Nash equilibria, both in theory [6, 27, 38] and practice [39]. In the two-person zero-sum case, large games are solved by pitting two regret-minimizing learning algorithms against each other, exploiting the fact that when both achieve a regret rate of $\epsilon/2$, their respective average strategies form an ϵ-Nash equilibrium [42]. For the game as described above, where the protagonist action is $\theta \in \Theta$ and the antagonist action is denoted σ_a, we imagine playing in rounds, where on round k the joint action is denoted by $\sigma^{(k)} = (\theta^{(k)}, \sigma_a^{(k)})$. Since the utility function for each player U^i for $i \in \{p, a\}$, is affine in their own action choice for any fixed action chosen by the other player, each faces an online convex optimization problem [41] (note that maximizing U^i is equivalent to minimizing $-U^i$; see also Appendix G). The **total regret** of a player, say the protagonist, is defined with respect to their utility function after K rounds as $R^p(\sigma^{(1)} \dots \sigma^{(K)}) = \max_{\theta \in \Theta} \sum_{k=1}^{K} U^p(\theta, \sigma_a^{(k)}) - U^p(\theta^{(k)}, \sigma_a^{(k)})$. (Nature can also be introduced to choose a random training example on each round, which simply requires the definition of regret to be expressed in terms of expectations over nature's choices.)

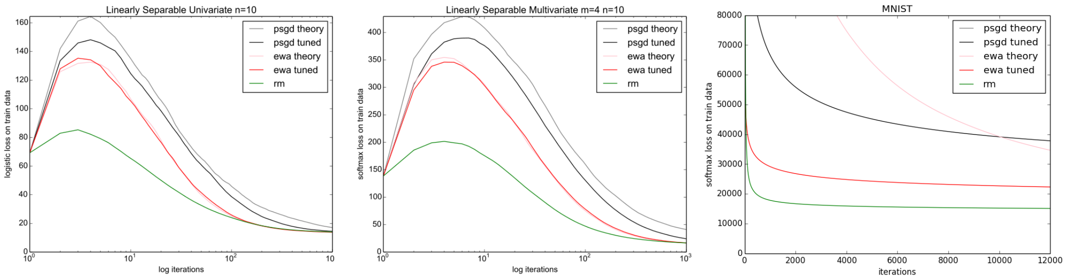
To accommodate regularization in the learning problem, we impose parameter constraints Θ. A particularly interesting case occurs when one defines $\Theta = \{\theta : \|\theta\|_1 \leq \beta\}$, since the L_1 ball constraint is equivalent to imposing L_1 regularization. There are two distinct advantages to L_1 regularization in this context. First, as is well known, L_1 encourages sparsity in the solution. Second, and much less appreciated, is the fact that *any* polytope constraint allows one to reduce the constrained online convex optimization problem to learning from expert advice over a finite number of experts [41]: Given a polytope Θ, define the **convex hull basis** $H(\Theta)$ to be a matrix whose columns are the vertices in Θ. An expert can then be assigned to each vertex in $H(\Theta)$, and an algorithm for learning from expert advice can then be applied by mapping its strategy on round k, $\rho^{(k)}$ (a probability distribution over the experts), back to an action choice in the original problem via $\theta^{(k)} = H(\Theta)\rho^{(k)}$, while the utility vector on round k, $u^{(k)}$, can be passed back to the experts via $H(\Theta)^\top u^{(k)}$ [41].

Since this reduction allows any method for learning from expert advice to be applied to L_1 constrained online convex optimization, we investigated whether alternative algorithms for supervised training might be uncovered. We considered two algorithms for learning from expert advice: the normalized **exponentiated weight algorithm** (EWA) [25, 36] (Algorithm 3); and **regret matching** (RM), a simpler method from the economics and game theory literature [13] (Algorithm 2). For supervised learning, these algorithms operate by using a stochastic sample of the gradient to perform their updates (outer loop Algorithm 1). EWA possesses superior regret bounds that demonstrate only a logarithmic dependence on the number of actions; however RM is simpler, hyperparameter-free, and still possesses reasonable regret bounds [10, 11]. Although exponentiated gradient methods have been applied to supervised learning [21, 36], we not aware of any previous attempt to apply regret matching to supervised training. We compared these to **projected stochastic gradient descent** (PSGD), which is the obvious modification of stochastic gradient descent (SGD) that retains a similar regret bound [8, 31] (Algorithm 4).

2.2 Evaluation

To investigate the utility of these methods for supervised learning, we conducted experiments on synthetic data and on the MNIST data set [23]. Note that PSGD and EWA have a step size parameter, $\eta^{(k)}$, that greatly affects their performance. The best regret bounds are achieved for step sizes of the form $\eta k^{-1/2}$ and $\eta \log(m) k^{-1/2}$ respectively [31]; we also tuned η to generate the best empirical results. Since the underlying optimization problems are convex, these experiments merely focus on the speed of convergence to a global minimum of the constrained training problem.

The first set of experiments considered synthetic problems. The data dimension was set to $m = 10$, and $T = 100$ training points were drawn from a standard multivariate Gaussian. For univariate prediction, a random hyperplane was chosen to label the data (hence the data was linearly separable, but not with a large margin). The *logistic* training loss achieved by the running average of the protagonist strategy $\bar{\theta}$ over the entire training set is plotted in Figure 1a. For multivariate prediction, a 4×10 target matrix, θ^*, was randomly generated to label training data by $\arg\max(\theta^* x_t)$. The training *softmax* loss achieved by the running average of the protagonist strategy $\bar{\theta}$ over the entire training set is shown in Figure 1b. The third experiment was conducted on MNIST, which is an $n = 10$ class problem over $m = 784$ dimensional inputs with $T = 60,000$ training examples, evidently not linearly separable. For this experiment, we used mini-batches of size 100. The training loss of the running average protagonist strategy $\bar{\theta}$ (single run) is shown in Figure 1c. The apparent effectiveness of RM in these experiments is a surprising outcome. Even after tuning η for both PSGD and EWA, they do not surpass the performance of RM, which is hyperparameter free. We did not anticipate this observation; the effectiveness of RM for supervised learning appears not to have been previously noticed. (We do not expect RM to be competitive in high dimensional *sparse* problems, since its regret bound has a square root and not a logarithmic dependence on n [10].)



(a) Logistic loss, synthetic data. (b) Softmax loss, synthetic data. (c) Softmax loss, MNIST data.

Figure 1: Training loss achieved by different no-regret algorithms. Subfigures (a) and (b) are averaged over 100 repeats, log scale x-axis. Subfigure (c) is averaged over 10 repeats (psgd theory off scale).

3 Deep Learning Games

A key contribution of this paper is to show how the problem of training a feedforward neural network with differentiable convex gates can be reduced to a game. A practical consequence of this reduction is that it suggests new approaches to training deep models that are inspired by methods that have recently proved successful for solving massive-scale games.

Feedforward Neural Network A feedforward neural network is defined by a directed acyclic graph with additional objects attached to the vertices and edges. The network architecture is specified by $N = (V, E, I, O, F)$, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, $I = \{i_1 \dots i_m\} \subset V$ is a set of input vertices, $O = \{o_1 \dots o_n\} \subset V$ is a set of output vertices, and $F = \{f_v : v \in V\}$ is a set of activation functions, where $f_v : \mathbb{R} \to \mathbb{R}$. The trainable parameters are given by $\theta : E \to \mathbb{R}$.

In the graph defined by $G = (V, E)$, a **path** $(v_1, ..., v_k)$ consists of a sequence of vertices such that $(v_j, v_{j+1}) \in E$ for all j. A **cycle** is a path where the first and last vertex are equal. We assume that G contains no cycles, the input vertices have no incoming edges (i.e. $(u, i) \notin E$ for all $i \in I$, $u \in V$), and the output vertices have no outgoing edges (i.e. $(o, v) \notin E$ for all $o \in O$, $v \in V$). A directed acyclic graph generates a partial order $\leq$ on the vertices where $u \leq v$ if and only if there is a path from u to v. For all $v \in V$, define $E_v = \{(u, u') \in E : u' = v\}$. The network is related to the training data by assuming $|I| = m$, the number of input vertices corresponds to the number of input features, and $|O| = n$, the number of output vertices corresponds to the number of output dimensions. It is a good idea (but not required) to have two additional bias inputs, whose corresponding input features are always set to 0 and 1, respectively, and have edges to all non-input nodes in the graph. Usually, the activation functions on input and output nodes are the identity, i.e. $f_v(x) = x$ for $v \in I \cup O$.

Given a training input $x_t \in \mathbb{R}^m$, the computation of the network N is expressed by a circuit value function c_t that assigns values to each vertex based on the partial order over vertices:

$$c_t(i_k, \theta) = f_{i_k}(x_{tk}) \text{ for } i_k \in I; \quad c_t(v, \theta) = f_v\big(\textstyle\sum_{u:(u,v)\in E} c_t(u, \theta)\theta(u, v)\big) \text{ for } v \in V - I. \tag{1}$$

Let $c_t(\mathbf{o}, \theta)$ denote the vector of values at the output vertices, i.e. $(c_t(\mathbf{o}, \theta))_k = c_t(o_k, \theta)$. Since each f_v is assumed differentiable, the output $c_t(\mathbf{o}, \theta)$ must also be differentiable with respect to θ.

When we wish to impose constraints on θ we assume the constraints factor over vertices, and are applied across the incoming edges to each vertex. That is, for each $v \in V - I$ the parameters θ restricted to E_v are required to be in a set $\Theta_v \subseteq \mathbf{R}^{E_v}$, and $\Theta = \prod_{v \in V - I} \Theta_v$. (We additionally assume each Θ_v satisfies constraint qualifications—see Appendix C—and can also alter the factorization requirement to allow more complex network architectures—see Appendix H). If $\Theta = \mathbf{R}^E$, we consider the network to be **unconstrained**. If Θ is bounded, we consider the network to be **bounded**.

DLP (Deep Learning Problem) Given a loss function $\ell(z, y)$ that is convex in the first argument satisfying $0 \leq \ell(z, y) < \infty$ for all $z \in \mathbb{R}^n$, define $\ell_t(z) = \ell(z, y_t)$ and $L_t(\theta) = \ell_t(c_t(\mathbf{o}, \theta))$. The training problem is to find a $\theta \in \Theta$ that minimizes $L(\theta) = T^{-1} \sum_{t=1}^{T} L_t(\theta)$.

DLG (Deep Learning Game) We define a one-shot simultaneous move game [40, page 9] with infinite action sets (Appendix E); we need to specify the players, action sets, and utility functions.

Players: The players consist of a protagonist p for each $v \in V - I$, an antagonist a, and a set of self-interested zannis s_v, one for each vertex $v \in V$.[2] *Actions:* The protagonist for vertex v chooses a parameter function $\theta_v \in \Theta_v$. The antagonist chooses a set of T vectors and scalars $\{a_t, b_t\}_{t=1}^{T}$, $a_t \in \mathbb{R}^n$, $b_t \in \mathbb{R}$, such that $a_t^\top z + b_t \leq \ell_t(z)$ for all $z \in \mathbb{R}^n$; that is, the antagonist chooses an affine minorant of the local loss for each training example. Each zanni s_v chooses a set of $2T$ scalars (q_{vt}, d_{vt}), $q_{vt} \in \mathbb{R}$, $d_{vt} \in \mathbb{R}$, such that $q_{vt} z + d_{vt} \leq f_v(z)$ for all $z \in \mathbb{R}$; that is, the zanni chooses an affine minorant of its local activation function f_v for each training example. All players make their action choice without knowledge of the other player's choice. *Utilities:* For a joint action $\sigma = (\theta, \{a_t, b_t\}, \{q_{vt}, d_{vt}\})$, the zannis' utilities are defined recursively following the parial order on vertices. First, for each $i \in I$ the utility for zanni s_i on training example t is $U_{it}^s(\sigma) = d_{it} + q_{it} x_{it}$, and for each $v \in V - I$ the utility for zanni s_v on example t is $U_{vt}^s(\sigma) = d_{vt} + q_{vt} \sum_{u:(u,v) \in E} U_{tu}^s(\sigma) \theta(u, v)$. The total utility for each zanni s_v is given by $U_v^s(\sigma) = \sum_{t=1}^{T} U_{vt}^s(\sigma)$ for $v \in V$. The utility for the antagonist a is then given by $U^a = T^{-1} \sum_{t=1}^{T} U_t^a$ where $U_t^a(\sigma) = b_t + \sum_{k=1}^{n} a_{kt} U_{o_k t}^s(\sigma)$. The utility for all protagonists are the same, $U^p(\sigma) = -U^a(\sigma)$. (This representation also allows for an equivalent game where nature selects an example t, tells the antagonist and the zannis, and then everyone plays their actions simultaneously.) The next lemma shows how the zannis and the antagonist can be expected to act.

Lemma 3 *Given a fixed protagonist action θ, there exists a unique joint action for all agents $\sigma = (\theta, \{a_t, b_t\}, \{q_{vt}, d_{vt}\})$ where the zannis and the antagonist are playing best responses to σ. Moreover, $U^p(\sigma) = -L(\theta)$, $\nabla_\theta U^p(\sigma) = -\nabla L(\theta)$, and given some protagonist at $v \in V - I$, if we hold all other agents' strategies fixed, $U^p(\sigma)$ is an affine function of the strategy of the protagonist at v. We define σ as the* ***joint action expansion for θ****.*

There is more detail in the appendix about the joint action expansion. However, the key point is that if the current cost and partial derivatives can be calculated for each parameter, one can construct the affine function for each agent. We will return to this in Section 3.1.

A **KKT point** is a point that satisfies the KKT conditions [18, 22]: roughly, that either it is a **critical point** (where the gradient is zero), or it is a point on the boundary of Θ where the gradient is pointing out of Θ "perpendicularly" (see Appendix C). We can now state the main theorem of the paper, showing a one to one relationship between KKT points and Nash equilibria.

Theorem 4 **(DLG Nash Equilibrium)** *The joint action $\sigma = (\theta, \{a_t, b_t\}, \{q_{vt}, d_{vt}\})$ is a Nash equilibrium of the DLG iff it is the joint action expansion for θ and θ is a KKT point of the DLP.*

Corollary 5 *If the network is unbounded, the joint action $\sigma = (\theta, \{a_t, b_t\}, \{q_{vt}, d_{vt}\})$ is a Nash equilibrium of the DLG iff it is the joint action expansion for θ and θ is a critical point of the DLP.*

Finally we note that sometimes we need to add constraints between edges incident on different nodes. For example, in a convolutional neural network, one will have edges $e = \{u, v\}$ and $e' = \{u', v'\}$ such that there is a constraint $\theta_e = \theta_{e'}$ (see Appendix H). In game theory, if two agents act simultaneously it is difficult to have one agent's viable actions depend on another agent's action. Therefore, if parameters are constrained in this manner, it is better to have one agent control both. The appendix (beginning with Appendix B) extends our model and theory to handle such parameter tying, which allows us to handle both convolutional networks and non-convex activation functions (Appendix I). Our theory does not apply to non-smooth activation functions, however (e.g. ReLU gates), but these can be approximated arbitrarily closely by differentiable activations.

3.1 Learning Algorithms

Characterizing the deep learning problem as a game motivates the consideration of equilibrium finding methods as potential training algorithms. Given the previous reduction to expert algorithms, we will consider the use of the L_1 ball constraint $\Theta_v = \{\theta_v : \|\theta_v\|_1 \leq \beta\}$ at each vertex v. For deep learning, we have investigated a simple approach by training independent protagonist agents at each vertex against a best response antagonist and best response zannis [17]. In this case, it is possible

[2] Nomenclature explanation: Protagonists nominally strive toward a common goal, but their actions can interfere with one another. Zannis are traditionally considered servants, but their motivations are not perfectly aligned with the protagonists. The antagonist is diametrically opposed to the protagonists.

Algorithm 1 Main Loop

```
On round $k$, observe some $x_t$ (or mini batch)
Antagonist and zannis choose best responses
    which ensures $\nabla U_v^p(\theta_v) = -\nabla L(\theta_v^{(k)})$
$g_v^{(k)} \leftarrow \nabla U_v^p(\theta_v)$
Apply update to $r_v^{(k)}$, $\rho_v^{(k)}$ and $\theta_v^{(k)}$ $\forall v \in V$
```

Algorithm 2 Regret Matching (RM)

```
$r_v^{(k+1)} \leftarrow r_v^{(k)} + H(\Theta_v)^\top g_v^{(k)} - \rho_v^{(k)\top} H(\Theta_v)^\top g_v^{(k)}$
$\rho_v^{(k+1)} \leftarrow \left(r_v^{(k+1)}\right)_+ / \left(\mathbf{1}^\top \left(r_v^{(k+1)}\right)_+\right)$
$\theta_v^{(k+1)} \leftarrow H(\Theta_v)\rho_v^{(k+1)}$
```

Algorithm 3 Exp. Weighted Average (EWA)

```
$r_v^{(k+1)} \leftarrow r_v^{(k)} + \eta^{(k)} H(\Theta_v)^\top g_v^{(k)}$
$\rho_v^{(k+1)} \leftarrow \exp(r_v^{(k+1)})/(\mathbf{1}^\top \exp(r_v^{(k+1)}))$
$\theta_v^{(k+1)} \leftarrow H(\Theta_v)\rho_v^{(k+1)}$
```

Algorithm 4 Projected SGD

```
$r_v^{(k+1)} \leftarrow r_v^{(k)} + \eta^{(k)} H(\Theta_v)^\top g_v^{(k)}$
$\rho_v^{(k+1)} \leftarrow L_2\text{\_project\_to\_simplex}(r_v^{(k+1)})$
$\theta_v^{(k+1)} \leftarrow H(\Theta_v)\rho_v^{(k+1)}$
```

to devise interesting and novel learning strategies based on the algorithms for learning from expert advice. Since the optimization problem is no longer convex in a local protagonist action θ_v, we do not expect convergence to a joint, globally optimal strategy among protagonists. Nevertheless, one can develop a generic approach for using the game to generate a learning algorithm.

Algorithm Outline On each round, nature chooses a random training example (or mini-batch). For each $v \in V$, each protagonist v selects her actions $\theta_v \in \Theta_v$ deterministically. The antagonist and zannis then select their actions, which are best responses to the θ_v and to each other.[3] The protagonist utilities U_v^p are then calculated. Given the zanni and antagonist choices, U_v^p is affine in the protagonist's action, and also by Lemma 3 for all $e \in E_v$, we have $\frac{\partial L^t}{\partial w_e} = -\frac{\partial U_v^p(\theta_v)}{\partial w_e}$. Each protagonist $v \in V$ then observes their utility and uses this to update their strategy. See Algorithm 1 for the general loop, and Algorithms 2, 3 and 4 for specific updates.

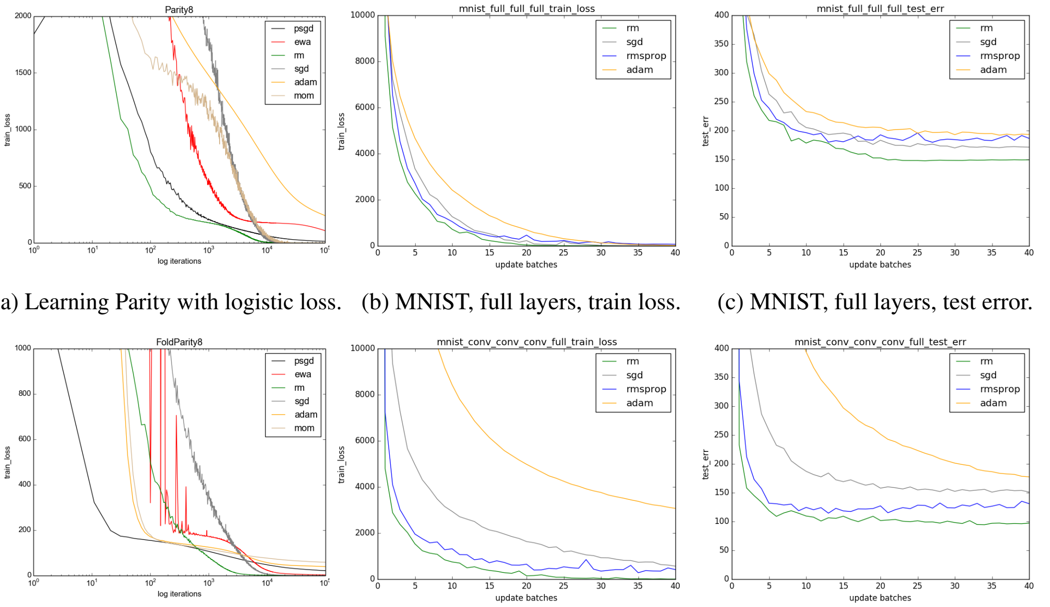
Given the characterization developed previously, we know that a Nash equilibrium will correspond to a critical point in the training problem (which is almost certain to be a local minimum rather than a saddle point [24]). It is interesting to note that the usual process of backpropagating the sampled (sub)gradients corresponds to computing the best response actions for the zannis and the antagonist, which then yields the resulting affine utility for the protagonists.

3.2 Experimental Evaluation

We conducted a set of experiments to investigate the plausibility of applying expert algorithms at each vertex in a feedforward neural network. For comparison, we considered current methods for training deep models, including SGD [3], SGD with momentum [37], RMSprop, Adagrad [7], and Adam [20]. Since none of these impose constraints, they technically solve an easier optimization problem, but they are also un-regularized and therefore might exhibit weaker generalization. We tuned the step size parameter for each comparison method on each problem. For the expert algorithms, RM, EWA and PSGD, we found that EWA and PSGD were not competitive, even after tuning their step sizes. For RM, we initially found that it learned too quickly, with the top layers of the model becoming sparse; however, we discovered that RM works remarkably well simply by initializing the cumulative regret vectors $r_v^{(0)}$ with random values drawn from a Gaussian with large standard deviation σ.

As a sanity check, we first conducted experiments on synthetic combinatorial problems: "parity", defined by $y = x_1 \oplus \cdots \oplus x_m$ and "folded parity", defined by $y = (x_1 \wedge x_2) \oplus \cdots \oplus (x_{m-1} \wedge x_m)$ [30]. Parity cannot be approximated by a single-layer model but is representable with a single hidden layer of linear threshold gates [12], while folded parity is known to be not representable by a (small weights) linear threshold circuit with only a single hidden layer; at least two hidden layers are required [30]. For parity we trained a m-$4m$-1 architecture, and for folded parity we trained a m-$4m$-$4m$-1 architecture, both fully connected, $m = 8$. Here we chose the L_1 constraint bound to be $\beta = 10$ and the initialization scale as $\sigma = 100$. For the nonlinear activation functions we used a smooth

[3] Conceptually, each zanni has a copy of the algorithm of each protagonist and an algorithm for selecting a joint action for all antagonists and zannis, and thus do not technically depend upon θ_v. In practice, these multiple copies are unnecessary, and one merely calculates $\theta_v \in \Theta_v$ first.



(a) Learning Parity with logistic loss. (b) MNIST, full layers, train loss. (c) MNIST, full layers, test error.

(d) Folded Parity, logistic loss. (e) MNIST, convolutional, train loss. (f) MNIST, convolutional, test error.

Figure 2: Experimental results. (a) Parity, m-$4m$-1 architecture, 100 repeats. (d) Folded parity, m-$4m$-$4m$-1 architecture, 100 repeats. (b) and (c): MNIST, 784-1024-1024-10 architecture, 10 repeats. (e) and (f): MNIST, 28×28-$c(5\times5, 64)$-$c(5\times5, 64)$-$c(5\times5, 64)$-10 architecture, 10 repeats.

approximation of the standard ReLU gate $f_v(x) = \tau \log(1 + e^{x/\tau})$ with $\tau = 0.5$. The results shown in Figure 2a and Figure 2d confirm that RM performs competitively, even when producing models with sparsity, top to bottom, of 18% and 13% for parity, and 27%, 19% and 21% for folded parity.

We next conducted a few experiments on MNIST data. The first experiment used a fully connected 784-1024-1024-10 architecture, where RM was run with $\beta = 30$ and initialization scales $(\sigma_1, \sigma_2, \sigma_3) = (50, 200, 50)$. The second experiment was run with a convolutional architecture 28×28-c(5×5, 64)-c(5×5, 64)-c(5×5, 64)-10 (convolution windows 5×5 with depth 64), where RM was run with $(\beta_1, \beta_2, \beta_3, \beta_4) = (30, 30, 30, 10)$ and initialization scales $\sigma = 500$. The mini-batch size was 100, and the x-axis in the plots give results after each "update" batch of 600 mini-batches (i.e. one epoch over the training data). The training loss and test loss are shown in Figures 2b, 2c, 2e and 2f, showing the evolution of the training loss and test misclassification errors. We dropped all but SGD, Adam, RMSprop and RM here, since these seemed to dominate the other methods in our experiments. It is surprising that RM can demonstrate convergence rates that are competitive with tuned RMSprop, and even outperforms methods like SGD and Adam that are routinely used in practice. An even more interesting finding is that the solutions found by RM were sparse while achieving lower test misclassification errors than standard deep learning methods. In particular, in the fully connected case, the final solution produced by RM zeroed out 32%, 26% and 63% of the parameter matrices (from the input to the output layer) respectively. For the convolutional case, RM zeroed out 29%, 27%, 28% and 43% of the parameter matrices respectively. Regarding run times, we observed that our Tensorflow implementation of RM was only 7% slower than RMSProp on the convolutional architecture, but 85% slower in the fully connected case.

4 Related Work

There are several works that consider using regret minimization to solve offline optimization problems. Once stochastic gradient descent was connected to regret minimization in [5], a series of papers followed [29, 28, 34]. Two popular approaches are currently Adagrad [7] and traditional stochastic gradient descent. The theme of simplifying the loss is very common: it appears in batch gradient and incremental gradient approaches [26] as the majorization-minimization family of algorithms. In the regret minimization literature, the idea of simplifying the class of losses by choosing a minimizer from a particular family of functions first appeared in [41], and has since been further developed.

By contrast, the history of using games for optimization has a much shorter history. It has been shown that a game between people can be used to solve optimal coloring [19]. There is also a history of using regret minimization in games: of interest is [42] that decomposes a single agent into multiple agents, providing some inspiration for this paper. In the context of deep networks, a paper of interest connects brain processes to prediction markets [1]. However, the closest work appears to be the recent manuscript [2] that also poses the optimization of a deep network as a game. Although the games described there are similar, unlike [2], we focus on differentiable activation functions, and define agents with different information and motivations. Importantly, [2] does not characterize all the Nash equilibria in the game proposed. We discuss these issues in more detail in Appendix J.

5 Conclusion

We have investigated a reduction of deep learning to game playing that allowed a bijection between KKT points and Nash equilibria. One of the novel algorithms considered for supervised learning, regret matching, appears to provide a competitive alternative that has the additional benefit of achieving sparsity without unduly sacrificing speed or accuracy. It will be interesting to investigate alternative training heuristics for deep games, and whether similar successes can be achieved on larger deep models or recurrent models.

References

[1] D. Balduzzi. Cortical prediction markets. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, pages 1265–1272, 2014.

[2] D. Balduzzi. Deep online convex optimization using gated games. http://arxiv.org/abs/1604.01952, 2016.

[3] L. Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade - Second Edition*, pages 421–436. 2012.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge U. Press, 2004.

[5] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, September 2004.

[6] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[7] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[8] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *Inter. Conf. on Machine Learning*, pages 272–279, 2008.

[9] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

[10] G. Gordon. No-Regret algorithms for structured prediction problems. Technical Report CMU-CALD-05-112, Carnegie Mellon University, 2005.

[11] G. Gordon. No-regret algorithms for online convex programs. In *NIPS 19*, 2006.

[12] A. Hajnal. Threshold circuits of bounded depth. *JCSS*, 46(2):129–154, 1993.

[13] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[14] S. Hart and A. Mas-Colell. Regret-based continuous-time dynamics. *Games and Economic Behavior*, pages 375–394, 2003.

[15] K. Hoeffgen, H. Simon, and K. Van Horn. Robust trainability of single neurons. *JCSS*, 52(2):114–125, 1995.

[16] J. Hofbauer and W. H. Sandholm. On the global convergence of stochastic fictitious play. *Econometrica*, 70(6):2265–2294, 2002.

[17] M. Johanson, N. Bard, N. Burch, and M. Bowling. Finding optimal abstract strategies in extensive form games. In *AAAI Conference on Artificial Intelligence*, pages 1371–1379, 2012.

[18] W. Karush. Minima of functions of several variables with inequalities as side constraints. Master's thesis, Univ. of Chicago, Chicago, Illinois, 1939.

[19] M. Kearns, S. Suri, and N. Montfort. An experimental study of the coloring problem on human subject networks. *Science*, 313:824–827, 2006.

[20] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[21] J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

[22] H. Kuhn and A. Tucker. Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium*, pages 481–492. University of California Press, 1951.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[24] J. Lee, M. Simchowitz, M. Jordan, and B. Recht. Gradient Descent Only Converges to Minimizers. In *29th Annual Conference on Learning Theory*, volume 49, 2016.

[25] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

[26] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[27] A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems 26*, pages 3066–3074, 2013.

[28] N. Ratliff, D. Bagnell, and M. Zinkevich. Subgradient methods for structured prediction. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*, 2007.

[29] N. Ratliff, J. A. Bagnell, and M. Zinkevich. Maximum margin planning. In *Twenty Second International Conference on Machine Learning (ICML-06)*, 2006.

[30] A. Razborov. On small depth threshold circuits. In *Algorithm Theory (SWAT 92)*, 1992.

[31] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[32] S. Shalev-Shwartz and Y. Singer. Convex repeated games and Fenchel duality. In *NIPS 19*, 2006.

[33] S. Shalev-Shwartz and Y. Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.

[34] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

[35] M. Slater. Lagrange multipliers revisited: A contribution to nonlinear programming. 1950.

[36] N. Srinivasan, V. Ravichandran, K. Chan, J. Vidhya, S. Ramakirishnan, and S. Krishnan. Exponentiated backpropagation algorithm for multilayer feedforward neural networks. In *ICONIP*, volume 1, 2002.

[37] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings ICML*, pages 1139–1147, 2013.

[38] V. Syrgkanis, A. Agarwal, H. Luo, and R. Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems 28*, pages 2971–2979, 2015.

[39] O. Tammelin, N. Burch, M. Johanson, and M. Bowling. Solving heads-up limit Texas hold'em. In *International Joint Conference on Artificial Intelligence, IJCAI*, pages 645–652, 2015.

[40] V. Vazirani, N. Nisan, T. Roughgarden, and É Tardos. *Algorithmic Game Theory*. Cambridge Press, 2007.

[41] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Twentieth International Conference on Machine Learning*, 2003.

[42] M. Zinkevich, M. Bowling, M. Johanson, and C. Piccione. Regret minimization in games with incomplete information. In *NIPS*, 2007.

A Proofs for Section 2 (One-layer Case)

We assume that ℓ is convex and differentiable in its first argument.

Fact 6 (OLP Optimality) *Since each ℓ_t in the definition of L is convex and differentiable, L is convex and differentiable, and a necessary and sufficient condition for $\theta^* \in \arg\min_\theta L(\theta)$ is $\nabla L(\theta^*) = 0$ [4, Equation 4.22]*

For the one-layer learning game, OLG, given the simple structure of the utility functions the Nash equilibria are easy to characterize.

Lemma 7 (OLG Nash Equilibrium) *The joint action $\sigma = (\theta, \{a_t, b_t\})$ is a Nash equilibrium of the OLG if and only if $\ell_t(\theta x_t) = a_t^\top \theta x_t + b_t$, $a_t = \nabla \ell_t(g)|_{g=\theta x_t}$ (antagonist best response), and $T^{-1} \sum_{t=1}^T a_t x_t^\top = 0$ (protagonist best response).*

Proof: We prove a Nash equilibrium satisfies the two conditions; the converse is similar. The first condition is easiest: since for any $g \in \mathbb{R}^n$, $\ell_t(g) \geq a_t^\top g + b_t$, $\ell_t(\theta x_t) = a_t^\top \theta x_t + b_t$ represents the highest possible utility for the antagonist. Define $z_t = \theta x_t$. However, since ℓ_t is convex and differentiable everywhere, there is exactly one affine function that equals ℓ_t at a point and is less than or equal everywhere else, namely $h(g) = \nabla \ell_t(z_t)(g - z_t) + \ell_t(z_t)$. Thus, $a_t = \nabla \ell_t(g)|_{g=\theta x_t}$.

Insofar as the protagonist is concerned, given a strategy a_t, b_t for the adversary, the protagonist's utility is affine. Specifically, it is $U^p = -T^{-1} \sum_{t=1}^T (a_t)^\top \theta x_t + b_t$. Taking the gradient with respect to θ yields $-T^{-1} \sum_{t=1}^T a_t x_t^\top$, and setting it to zero guarantees that the protagonist is playing a best response. ■

Lemma 8 *For a protagonist action θ, given the best response of the antagonist, $\nabla_\theta U^p = -\nabla L(\theta)$.*

Proof (of Theorem 1): (1) From the conditions stated in Lemma 7 we must have $L(\theta) = T^{-1} \sum_t a_t^\top \theta x_t + b_t$ and $a_t = \nabla \ell_t(g)|_{g=\theta x_t}$ by antagonist best response. By protagonist best response, $T^{-1} \sum_{t=1}^T a_t x_t^\top = 0$, so $T^{-1} \sum_{t=1}^T \nabla \ell_t(g)|_{g=\theta x_t} x_t^\top = 0$. Since $\nabla L(\theta) = T^{-1} \sum_{t=1}^T \nabla \ell_t(g)|_{g=\theta x_t} x_t^\top$, $\nabla L(\theta) = 0$, and therefore by [4, Equation 4.22], θ is a minimum.

(2) If $\nabla L(\theta^*) = 0$ then $\sum_{t=1}^T \nabla L_t(\theta^*) = 0$. Since $L_t(\theta^*) = \ell_t(\theta^* x_t)$, then $\nabla L_t(\theta^*) = (\nabla \ell_t(g)|_{g=\theta^* x_t}) x_t^\top$. Define $a_t = \nabla \ell_t(g)|_{g=\theta^* x_t}$, and $b_t = \ell_t(\theta^* x_t) - a_t^\top \theta^* x_t$, such that the antagonist is playing a best response to θ^*. Notice that:

$$0 = \nabla L(\theta^*) \tag{2}$$

$$= T^{-1} \sum_{t=1}^T \nabla L_t(\theta^*) \tag{3}$$

$$= T^{-1} \sum_{t=1}^T (\nabla \ell_t(g)|_{g=\theta^* x_t}) x_t^\top \tag{4}$$

$$= T^{-1} \sum_{t=1}^T a_t x_t^\top \tag{5}$$

Thus, θ^* is also a best response, so $(\theta^*, \{a_t, b_t\})$ is an equilibrium. ■

For the constrained version of the one-layer neural network, we will temporarily assume that the constraint set Θ is a polytope. (The set of allowable constraints will be generalized throughout the remainder of the appendix, but linear constraints allow for a simple exposition to start.) Since a polytope is an intersection of a finite set of half-spaces, we can define such a Θ using a set J of affine functions, where $\theta \in \Theta$ iff for all $j \in J$, $j(\theta) \leq 0$.

To characterize the solutions of the constrained problem, we use the KKT conditions.

Fact 9 (OCP Optimality) *Since each ℓ_t in the definition of L is convex and differentiable, L is convex and differentiable, so necessary and sufficient conditions for $\theta^* \in \arg\min_{\theta \in \Theta} L(\theta)$ is that there exist*

$\{\mu_j\}_{j\in J}$ such that for all $j \in J$, $\mu_j \geq 0$, $\mu_j j(\theta^*) = 0$, $j(\theta^*) \leq 0$, and: $\sum_{j\in J} \mu_j j(\theta^*) = -\nabla L(\theta^*)$. [4, p. 244].

Lemma 10 (OCG Nash Equilibrium) *The joint action $\sigma = (\theta, \{a_t, b_t\})$ is a Nash equilibrium of the OLG if and only if $\ell_t(\theta x_t) = a_t^\top \theta x_t + b_t$, $a_t = \nabla \ell_t(g)|_{g=\theta x_t}$ (antagonist best response), and there exist $\{\mu_j\}_{j\in J}$ such that for all $j \in J$, $\mu_j \geq 0$, $\mu_j j(\theta) = 0$, $j(\theta) \leq 0$, and $\sum_{j\in J} \mu_j j(\theta^*) = -T^{-1} \sum_{t=1}^T a_t x_t^\top$ (protagonist best response).*

Proof: The antagonist best response proof is nearly identical to the proof of Lemma 7. The protagonist best response leverages that the gradient of U^p with respect to θ^* is $-T^{-1} \sum_{t=1}^T a_t x_t^\top$, and then leverages the KKT conditions for a maximum value. Since U^p is an affine function with respect to θ, a point satisfying the KKT conditions is a global maximum (Fact 9), implying it is a best response for the protagonist. Again, we leave proving the converse as an exercise. ■

Proof (of Theorem 2): (1) Using the same argument as the proof of (1) in Theorem 1, we can argue that $\nabla L(\theta^*) = -\nabla U^p$, as the antagonist best response constraints are the same as before. Using the protagonist best response constraints, we get that there exist $\{\mu_j\}_{j\in J}$ such that for all $j \in J$, $\mu_j \geq 0$, $\mu_j j(\theta^*) = 0$, $j(\theta^*) \leq 0$, and $\sum_{j\in J} \mu_j j(\theta^*) = -T^{-1} \sum_{t=1}^T a_t x_t^\top = \nabla U^p$. Thus:

$$\sum_{j\in J} \mu_j j(\theta^*) = -\nabla L(\theta^*) \tag{6}$$

Which are the KKT conditions from Fact 9, so θ^* is globally optimal.

(2) Assume that θ^* is an optimal solution. Using the same argument as the proof of (2) in Theorem 1, we construct a_t and b_t in the exact same way, as the antagonist best response property is the same as before, and as easily satisfied. This also implies that $\nabla L(\theta^*) = -\nabla U^p$. Using Fact 9, there exist $\{\mu_j\}_{j\in J}$ such that for all $j \in J$, $\mu_j \geq 0$, $\mu_j j(\theta^*) = 0$, and: $\sum_{j\in J} \mu_j j(\theta^*) = -\nabla L(\theta^*)$. So, $\sum_{j\in J} \mu_j j(\theta^*) = \nabla U^p = -T^{-1} \sum_{t=1}^T a_t x_t^\top$. Therefore, the protagonist best response property holds, and (θ^*, a_t, b_t) is a Nash equilibrium. ■

B Groups of Nodes

To handle two important extensions of feedforward neural networks (e.g. convolutional neural networks (Appendix H) and nonconvex activation functions (Appendix I)), we will need to extend the basic neural network model from the main body of the paper to consider constraints that couple the parameters defined on different edges. For example, in a convolutional neural network, you may have two edges $e = \{u, v\}$ and $e' = \{u', v'\}$ where there is a constraint that $\theta_e = \theta_{e'}$ (see Appendix H). In game theory, if two agents act simultaneously, it is difficult to have one agent's viable actions dependent upon the other agent's action. Thus, if two parameters are jointly constrained, it is best to have one agent control both parameters. Therefore, throughout the remainder of the appendix we will use a generalization of the model described in the body of the paper.

Define a partition P of $V - I$, where for each $\rho \in P$, $E_\rho = \cup_{v\in\rho} E_v$. Also define $\Theta_\rho \subseteq \mathbf{R}^{E_\rho}$, and $\Theta = \prod_{\rho\in P} \Theta_\rho$. An important constraint is that for any $\rho \in P$, and for any $u, v \in \rho$, if $u \leq v$ or $v \leq u$, then $u = v$. We leave the zannis' and the adversaries' action spaces unchanged (i.e., there is still one zanni per node), but each protagonist controls a partition of nodes. Notice that this is a strict generalization of the earlier model, because one could always define the discrete partition where each node is its own partition.

C Best Response and KKT Conditions

In this paper, we must analyze partial problems (related to best response in the game) of the form:

Partial problem at $\boldsymbol{\rho \in P}$: For an affine function $u : \mathbb{R}^{E_\rho} \to \mathbf{R}$, find $\operatorname{argmax}_{\theta_\rho\in\Theta_\rho} u(\theta_\rho)$.

For each $\rho \in P$, we will define $H_\rho \subseteq \mathbb{R}^{\mathbb{R}^{E_\rho}}$ and $J_\rho \subseteq \mathbb{R}^{\mathbb{R}^{E_\rho}}$ to be finite sets of continuous, differentiable functions. Then, we can define Θ_ρ to be the set of all $\theta_\rho \in \mathbf{R}^{E_\rho}$ where for all $h \in H_\rho$,

$h(\theta_\rho) = 0$, and for all $j \in J_\rho$, $j(\theta_\rho) \leq 0$. Before we look at the KKT conditions for the partial problem, we define two variations of constraint qualification:

1. **Partial affine constraint qualification:** For all $\rho \in P$, all $h \in H_\rho$ are affine and all $j \in J_\rho$ are affine.
2. **Partial Slater's constraint qualification:** For all $\rho \in P$, all $h \in H_\rho$ are affine and all $j \in J_\rho$ are convex, and there exists a $\theta_\rho \in \Theta_\rho$ where for all $j \in J_\rho$, $j(\theta_\rho) < 0$.

One classic constraint is $\sum_{e\in E_v} |\theta_e| \leq 1$, a bound on the L1 norm of the parameters of a particular vertex. This can be written as a set of linear inequalities (i.e. affine functions in J_ρ). Another is $\sum_{e\in E_v} (\theta_e)^2 \leq 1$, a bound on the L2 norm of the parameters of a particular vertex. This can be written as a convex constraint (i.e. $j(\theta_p) = \sum_{e\in E_v} (\theta_e)^2 - 1$).

We will define θ_ρ to be a **KKT point for a partial problem at $\boldsymbol{\rho \in P}$** if $\theta_\rho \in \Theta_\rho$ and there exists KKT multipliers $\mu_j \geq 0$ and $\lambda_h \in \mathbb{R}$ such that:

$$\nabla u(\theta_\rho) = \sum_{j\in J_\rho} \mu_j \nabla j(\theta_\rho) + \sum_{h\in H_\rho} \lambda_h \nabla h(\theta_\rho) \tag{7}$$

$$\mu_j j(\theta_\rho) = 0 \text{ for all } j \in J_\rho \tag{8}$$

In other words, the gradient points directly out of the feasible Θ_ρ.

Theorem 11 *[18, 22, 35, 4] Given either the partial affine constraint qualification, or the partial Slater's constraint qualification, any global minimum is a KKT point, and any KKT point is a global minimum.*

Notice that for the partial problem, we have assumed that the utility is an affine function, otherwise a KKT point would not necessarily be a global minimum. We will not make the same assumption for the full problem.

D Deep Learning and KKT Conditions

Now we will switch and consider the deep learning problem. In the deep learning problem, we want to find a **global minimum** $\theta^* \in \Theta$, such that for all $\theta \in \Theta$, $L(\theta^*) \leq L(\theta)$. This global minimum does not necessarily exist, nor is it necessarily unique. We can also define a distance function over $\mathbb{R}^E$, where for all $\theta, \bar{\theta} \in \Theta$, $d(\theta, \bar{\theta}) = \left(\sum_{e\in E} (\theta(e) - \bar{\theta}(e))^2\right)^{1/2}$. Define $N \subseteq \Theta$ to be a **neighborhood** of θ if there exists an $r > 0$ such that for all $\bar{\theta} \in \Theta$, $d(\theta, \bar{\theta}) < r$. θ is a **local minimum** if there exists a neighborhood N of θ such that for all $\bar{\theta} \in N$, $L(\theta) \leq L(\bar{\theta})$. Notice that a global minimum is a local minimum.

We will define $\theta \in \mathbb{R}^E$ to be a **KKT point** if $\theta \in \Theta$ and there exists KKT multipliers $\mu_j \geq 0$ and $\lambda_h \in \mathbb{R}$ such that:

$$-\nabla L(\theta) = \sum_{j\in J} \mu_j \nabla j(\theta) + \sum_{h\in H} \lambda_h \nabla h(\theta) \tag{9}$$

$$\mu_j j(\theta) = 0 \text{ for all } j \in J \tag{10}$$

In other words, the opposite of the gradient points directly out of the feasible Θ (this is a minimization problem, not a maximization problem). The KKT conditions explore properties for a point $\theta \in \Theta$ that are necessary (but not sufficient) for it to be a local minimum of some function L. They specify Θ by giving a set of constraints which must hold for all $\theta \in \Theta$.

For all $\rho \in P$, define $\Pi_\rho : \mathbb{R}^E \to \mathbb{R}^{E_\rho}$ such that for all $\theta \in \mathbf{R}^E$, for all $e \in E_\rho$, $(\Pi_\rho(\theta))_e = \theta_e$. We can define $H = \bigcup_{\rho\in P} \{h \circ \Pi_\rho\}_{h\in H_\rho}$ and $J = \bigcup_{\rho\in P} \{j \circ \Pi_\rho\}_{j\in J_\rho}$. Note that a point $\theta \in \mathbf{R}^E$ is in $\Theta = \prod_{\rho\in P} \Theta_\rho$ if and only if, for all $h \in H$, $h(\theta) = 0$, and for all $j \in J$, $j(\theta) \leq 0$.

1. **Full affine constraint qualification:** For all $\rho \in P$, all $h \in H$ are affine and all $j \in J$ are affine.
2. **Full Slater's Constraint Qualification:** For H, J, for all $\theta \in \Theta$, all $h \in H$ are affine, all $j \in J$ are convex, and there exists a $\theta \in \Theta$ where for all $j \in J$, $j(\theta) < 0$.

Theorem 12 *[18, 22, 35]Given the full affine constraint qualification or the full Slater's constraint qualification, any local minimum (and thus, the global minimum) is a KKT point.*

The converse is not true. For example, saddle points can be KKT points as well.

Lemma 13 *The partial affine constraint qualification implies the full affine constraint qualification.*

Proof: For any $\rho \in P$, if $f : \mathbf{R}^{E_\rho} \to \mathbf{R}$ is an affine function, then $f \circ \Pi_\rho$ is an affine function. Thus, for all $\rho \in P$, for all $h \in H_\rho$, $h \circ \Pi_\rho$ is an affine function, so all $h \in H$ are affine. Similiarly, all $j \in J$ are affine. ■

Lemma 14 *The partial Slater's Constraint Qualification implies the full Slater's constraint qualification.*

Proof: As in the proof of Lemma 13, since H_ρ is a set of affine functions, H is a set of affine functions. For each $\rho \in P$, there exists a $\theta_\rho \in \Theta_\rho$ where for all $j \in J_\rho$, $j(\theta_\rho) < 0$. If we define $\theta \in \mathbb{R}^E$ such that for all $\rho \in P$, $\Pi_\rho(\theta) = \theta_\rho$, then for all $j \in J$, $j(\theta) < 0$. Finally, for every $\rho \in P$, for every $j \in J_\rho$, since j is convex and Π_ρ is linear, $j \circ \Pi_\rho$ is convex. ■

E A Simultaneous Move Game

At a high level, in a simultaneous move game[40] there is:

1. a set of players N
2. a set (finite or infinite) of actions for each player Σ_i. A joint action set $\Sigma = \prod_{i \in N} \Sigma_i$.
3. a utility function for each player $u_i : \Sigma \to \mathbb{R}$.

For any $i \in N$, define $\Sigma_{-i} = \prod_{j \in N \setminus i} \Sigma_i$. Given $\mathbf{a} \in \Sigma$, we can write $\sigma_{-i} \in \Sigma_{-i}$ where for all $j \in N \setminus i$, $(\sigma_{-i})_j = \sigma_j$. Thus, for $\sigma_{-i} \in \Sigma_{-i}$ and $\sigma_i \in \Sigma_i$, we can define $\sigma_{-i} \circ \sigma_i \in \Sigma$, where $(\sigma_{-i} \circ \sigma_i)_i = \sigma_i$ and for any $j \in N \setminus i$, $(\sigma_{-i} \circ \sigma_i)_j = (\sigma_{-i})_j$.

A strategy $\sigma_i^* \in \Sigma_i$ is a **best response** to $\sigma_{-i} \in \Sigma_{-i}$ if for all $\sigma_i \in \Sigma_i$, $u_i(\sigma_{-i} \circ \sigma_i^*) \geq u_i(\sigma_{-i} \circ \sigma_i^*)$. A strategy $\sigma_i^* \in \Sigma_i$ is also called a best response to $\mathbf{a} \in \Sigma$ if it is a best response to σ_{-i}. A joint action $\mathbf{a}^* \in \Sigma$ is a **Nash equilibrium** if for all $i \in N$, σ_i^* is a best response to σ_{-i}^*.

F Reasonable Actions, Nash Equilibria

Given a joint action for the deep learning game $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$, and some $v \in V$, if f_v is convex and differentiable, define the zanni at v to be **reasonable** for $\mathbf{a}$ if for all $t \in \{1 \ldots T\}$, $q_{v,t} = f'_{v,t}(\sum_{u:(u,v) \in E} c_t(u, \theta)\theta(u, v))$, and $f_v(\sum_{u:(u,v) \in E} c_t(u, \theta)\theta(u, v))) = d_{v,t} + q_{v,t}(\sum_{u:(u,v) \in E} c_t(u, \theta)\theta(u, v))$. In other words, the values and the derivatives of f_v and $d_{v,t} + q_{v,t}x$ match for the activation energies present in the graph.

If the loss l is convex and partially differentiable in the first term, then the adversary is **reasonable** if for all $t \in \{1 \ldots T\}$, $a_t^\top c_t(\mathbf{o}, \theta) + b_t = l_t(c_t(\mathbf{o}, \theta))$ and $a_t = \nabla l_t(z)|_{z = c_t(\mathbf{o}, \theta)}$.

It is straightforward to think about strong induction over a partially ordered finite set.

Fact 15 *Given a finite set S, a partial ordering $\leq$ over S, and a set $X \subseteq S$, then if for all $s' \in S$, $\{s' \in S : s' < s\} \subseteq X \Rightarrow s \in X$, then $X = S$.*

Note that in strong induction, the base case is just a case $s \in S$ where $\{s' \in S : s' < s\} = \emptyset$.

Lemma 16 *Assume that for all $v \in V$, f_v is convex and differentiable. Assume $\leq$ is the partial order generated by the directed acyclic graph in the deep network. For any $v \in V$, given a joint action $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$ where for all $u \leq v$, the zanni at u is reasonable for* $\mathbf{a}$, *then $U_{tv}^s(\mathbf{a}) = c_t(v, \theta)$.*

Proof: Define $U \subseteq V$ to be the set of all vertices $u \in V$ where $u \leq v$. Define $R \subseteq U$ to be the set of nodes v where $U^s_{tv}(\mathbf{a}) = c_t(v, \theta)$. We can use the partial order of the graph as a partial order over U to prove recursively that $R = U$.

Then, we can prove by strong recursion on this total order that $U^s_{tu}(\mathbf{a}) = c_t(u, \theta)$ if for all $u' < u$, $U^s_{tu'}(\mathbf{a}) = c_t(u', \theta)$.

1. For any $u \in I$ (i.e. the base case), $U^s_{tu}(\mathbf{a}) = d_{u,t} + q_{u,t}(x_{t,u})$, and since the zanni at u is reasonable, $d_{u,t} + q_{u,t}(x_{t,u}) = f_v(x_{t,u}) = c_t(u, \theta)$.
2. For any $u \in U \backslash I$ (i.e. the inductive case), for all $(u', u) \in E$, $u' < u$, so $U^s_{tu'}(\mathbf{a}) = c_t(u', \theta)$, and thus $U^s_{tu}(\mathbf{a}) = d_{u,t} + q_{u,t}(\sum_{u':(u',u) \in E} c_t(u', \theta)\theta(u', u))$ Since the zanni is u is reasonable, $d_{u,t} + q_{v,t}(\sum_{u':(u',u) \in E} c_t(u', \theta)\theta(u', u)) = f_u(\sum_{u':(u',u) \in E} c_t(u', \theta)\theta(u', u)) = c_t(u, \theta)$.

■

Lemma 17 *Assume that for all $v \in V$, f_v is convex and differentiable, the loss l is convex and partially differentiable in the first term, and given a joint action $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$ where all zannis and the adversary are reasonable, then for any example t, $U^a_t(\mathbf{a}) = l_t(c_t(\mathbf{o}, \theta))$.*

Proof: The proof is analogous to the proof of Lemma 16. ■

Lemma 18 *Assume that for all $v \in V$, f_v is convex and differentiable. Assume $\leq$ is the partial order generated by the directed acyclic graph in the deep network. For any $v \in V$, given a joint action $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$ where for all $u \leq v$ (except possibly v), the zanni at u is reasonable, then unique best response for the zanni at v is to be reasonable.*

Proof: Since the zanni knows the example (or equivalently, chooses a different strategy based on the example), fix a specific example t. Define $z = x_{t,v}$ if $v \in I$, or $z = \sum_{u:(u,v) \in E} U^s_{tu}(\mathbf{a})\theta(u, v)$ otherwise. Then, the utility of the zanni at v is $d_{v,t} + q_{v,t}(z)$. First, observe that selecting $q_{t,v} = f'_v(z)$ and $d_{t,v} = f_v(z) - f'_v(z)z$ is a legal strategy for the zanni at v: because f_v is convex and differentiable, $f_v(x) \geq f'_v(z)(x - z) + f_v(z)$, so by definition $f_v(x) \geq d_{v,t} + q_{v,t}(x)$. Since for any legal strategy for the zanni at v, $f_v(z) \geq d_{v,t} + q_{v,t}(z)$, then this strategy maximizes utility for the zanni at v, because $f_v(z) = d_{v,t} + q_{v,t}(z)$. Moreover, since f_v is convex and differentiable, this affine function, which is equal to f_v at z and less than or equal to f_v everywhere else, is unique. Finally, note that if $v \in I$, $z = x_{t,v}$, and if $v \notin I$, from Lemma 16, we know that for all $\{u : (u, v) \in E\}$, $U^s_{tu}(\mathbf{a}) = c_t(u, \theta)$, so $z = \sum_{u:(u,v) \in E} c_t(u, \theta)\theta(u, v)$. ■

Thus, any time all the zannis are playing a best response, they are reasonable, and vice-versa. To complete the story, we consider the adversary.

Lemma 19 *Assume that for all $v \in V$, f_v is convex and differentiable, the loss l is convex and partially differentiable in the first term, and given a joint action $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$ where all zannis are reasonable, then the unique best response for the adversary is to be reasonable.*

Proof: The proof is analogous to Lemma 19. ■

The above lemmas state that it is easy to determine and reason about the best responses of the adversary and the zanni. Now, we go deeper into the analysis to reason about the protagonist.

Lemma 20 *Assume that for all $v \in V$, f_v is convex and differentiable, the loss l is convex and partially differentiable in the first term, and given a joint action $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$ where all zannis are reasonable, and the adversary is reasonable, and the protagonists play θ, then if U^p is the utility of the protagonists, then:*

$$\nabla_\theta U^p(\mathbf{a}) = -\nabla_\theta L(\theta) \tag{11}$$

Proof: First, we break apart U^p into U_t^p, where U_t^p is the utility of p conditional on nature selecting example t.

$$U_t^p(\mathbf{a}) = -U_t^a(\mathbf{a}) \tag{12}$$

$$= -b_t - \sum_{k=1}^{n} a_{t,k}(U_t^s(o_k)) \tag{13}$$

If we can prove $\nabla_\theta U_t^p(\mathbf{a}) = -\nabla_\theta l_t(c_t(\mathbf{o},\theta))$, the result follows quickly. Taking the partial derivative above, and relying on the lack of outgoing edges from o_k:

$$\frac{\partial U_t^p(\mathbf{a})}{\partial U_t^s(o_k)} = -a_{kt} \tag{14}$$

Since the adversary is reasonable, $a_{kt} = \frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(o_k,\theta)}$, and:

$$\frac{\partial U_t^p(\mathbf{a})}{\partial U_t^s(o_k)} = -\frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(o_k,\theta)} \tag{15}$$

Define $X \subseteq V$ to be the set of all $v \in V$ such that $\frac{\partial U_t^p(\mathbf{a})}{\partial U_{tv}^s(\mathbf{a})} = -\frac{\partial l_t(\mathbf{o},\theta)}{\partial c_t(v,\theta)}$. We consider the partial order $\leq$ over the vertices V in the deep network generated by the directed acyclic graph of the deep network: however we apply induction on the opposite partial order $\sqsubseteq$. We have shown $O \subseteq X$ above. We recursively show $X \supseteq V \backslash O$ below.

For some $v \in V \backslash O$, we want to show $v \in X$, and we assume that for all $u \in V$ where $u \sqsubset v$, $u \in X$. Notice that:

$$\frac{\partial U_t^p(\mathbf{a})}{\partial U_{tv}^s(\mathbf{a})} = \sum_{u:(v,u)\in E} \theta_{v,u} q_{u,t} \frac{\partial U_t^p(\mathbf{a})}{\partial U_{tu}^s(\mathbf{a})} \tag{16}$$

Since $(v,u) \in E$, $u \in V$, $u \sqsubset v$, and $u \neq v$, so by the inductive hypothesis:

$$\frac{\partial U_t^p(\mathbf{a})}{\partial U_{tv}^s(\mathbf{a})} = -\sum_{u:(v,u)\in E} \theta_{v,u} q_{u,t} \frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(u,\theta)} \tag{17}$$

Because $\mathbf{a}$ has a reasonable zanni at all $u \in V$, $q_{u,t} = f_u'\left(\sum_{u':(u',u)\in E} \theta(u',u) c_t(u',\theta)\right)$:

$$\frac{\partial U_t^p(\mathbf{a})}{\partial U_{tv}^s(\mathbf{a})} = -\sum_{u:(v,u)\in E} \theta_{v,u} f_u'\Big(\sum_{u':(u',u)\in E} \theta(u',u) c_t(u',\theta)\Big) \frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(u,\theta)} \tag{18}$$

$$\frac{\partial U_t^p(\mathbf{a})}{\partial U_{tv}^s(\mathbf{a})} = -\frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(v,\theta)} \tag{19}$$

Now we know for all $v \in V$, $\frac{\partial U_t^p(\mathbf{a})}{\partial U_t^s(v)} = -\frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(v,\theta)}$. We can now consider, for any $(u,v) \in E$, the partial derivative with respect to $\theta(u,v)$:

$$\frac{\partial U_t^p(\mathbf{a})}{\partial \theta(u,v)} = \frac{\partial U_t^p(\mathbf{a})}{\partial U_{tv}^s(\mathbf{a})} q_{v,t} U_{tu}^s(\mathbf{a}) \tag{20}$$

$$\frac{\partial U_t^p(\mathbf{a})}{\partial \theta(u,v)} = -\frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(v,\theta)} q_{v,t} U_{tu}^s(\mathbf{a}) \tag{21}$$

Because the zannis are reasonable, $q_{v,t} = f_v'(\sum_{u':(u',v)\in E} \theta(u',v) c_t(u',\theta))$ and $U_{tu}^s = c_t(u,\theta)$:

$$\frac{\partial U_t^p(\mathbf{a})}{\partial \theta(u,v)} = -\frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(v,\theta)} f_v'\Big(\sum_{u':(u',v)\in E} \theta(u',v) c_t(u',\theta)\Big) c_t(u,\theta) \tag{22}$$

Since $c_t(v,\theta) = f_v(\sum_{u':(u',v)\in E} \theta(u',v) c_t(u',\theta))$, $\frac{\partial c_t(v,\theta)}{\partial \theta(u,v)} = f_v'(\sum_{u':(u',v)\in E} \theta(u',v) c_t(u',\theta)) c_t(u,\theta)$, hence

$$\frac{\partial U_t^p(\mathbf{a})}{\partial \theta(u,v)} = -\frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial c_t(v,\theta)} \frac{\partial c_t(v,\theta)}{\partial \theta(u,v)} \tag{23}$$

$$\frac{\partial U_t^p(\mathbf{a})}{\partial \theta(u,v)} = -\frac{\partial l_t(c_t(\mathbf{o},\theta))}{\partial \theta(u,v)} \tag{24}$$

Averaging across examples yields the result. ■

For the deep network graph, define $P(u,v)$ to be the set of all paths from u to v, and for any path p, define $|p|$ to be the number of nodes in the path. Thus, for all $p \in P(u,v)$, $p_1 = u$ and $p_{|p|} = v$. The following lemma establishes the partial derivative with respect to $\theta(u,v)$. The key point of the lemma is that if $(u,v), (u',v') \in E_\rho$, then the partial derivative of $\theta(u,v)$ does not depend upon $\theta(u',v')$, and therefore if we restrict ourselves to modifying the weights in θ_ρ, U^p is affine.

Lemma 21

$$\frac{\partial U^p(\mathbf{a})}{\partial \theta(u,v)} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}\sum_{p\in P(v,o_k)} U^s_{tu}(\mathbf{a}) q_{t,p_{|p|}} a_{kt} \prod_{j=1}^{|p|-1} \theta(p_j,p_{j+1}) q_{t,p_j} \tag{25}$$

Proof: Notice that:

$$\frac{\partial U^p(\mathbf{a})}{\partial \theta(u,v)} = \sum_{t=1}^{T} U^s_{tu}(\mathbf{a}) q_{t,v} \frac{\partial U^p(\mathbf{a})}{\partial U^s_{tv}(\mathbf{a})} \tag{26}$$

Thus, the problem can be reduced to proving recursively, starting from O:

$$\frac{\partial U^p(\mathbf{a})}{\partial U^s_{tv}(\mathbf{a})} = -\frac{1}{T}\sum_{k=1}^{n} a_{kt} \sum_{p\in P(v,o_k)} \prod_{j=1}^{|p|-1} \theta(p_j,p_{j+1}) q_{t,p_{j+1}} \tag{27}$$

■

To clarify the decisions of the protagonists, for each $\rho \in P$, define $U^p_{\rho,\mathbf{a}} : \mathbf{R}^{E_\rho} \to \mathbf{R}$ such that $U^p_{\rho,\mathbf{a}}(\theta_\rho)$ is the utility of the protagonist at ρ if she unilaterally deviates from $\mathbf{a}$ to play θ_ρ.

Lemma 22 *$U^p_{\rho,\mathbf{a}}$ is an affine function.*

Proof: Fix a specific $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$. We can define $\mathbf{a}|_\rho : \Theta_\rho \to \Sigma$ such that for any $\tilde{\theta} \in \Theta_\rho$, $\mathbf{a}|_\rho(\tilde{\theta})$ is the same as $\mathbf{a}$ except the action of the protagonist at ρ is replaced by $\tilde{\theta}$. So:

$$U^p_{\rho,\mathbf{a}}(\tilde{\theta}) = U^p(\mathbf{a}|_\rho(\tilde{\theta})) \tag{28}$$

This tortured nomenclature allows us to say, for any $(u,v) \in E_\rho$:

$$\frac{\partial U^p_{\rho,\mathbf{a}}(\tilde{\theta})}{\partial \tilde{\theta}_{(u,v)}} = \frac{\partial U^p(\mathbf{a}|_\rho(\tilde{\theta}))}{\partial \tilde{\theta}_{(u,v)}} \tag{29}$$

From Lemma 21

$$\frac{\partial U^p(\mathbf{a}|_\rho(\tilde{\theta}))}{\partial \tilde{\theta}(u,v)} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}\sum_{p\in P(v,o_k)} U^s_{tu}(\mathbf{a}|_\rho(\tilde{\theta})) q_{t,p_{|p|}} a_{kt} \prod_{j=1}^{|p|-1} \theta(p_j,p_{j+1}) q_{t,p_j} \tag{30}$$

Thus, the function $U^p_{\rho,\mathbf{a}}$ is differentiable everywhere. Moreover, consider $U^s_{tu}(\mathbf{a}|_\rho(\tilde{\theta}))$. Notice $v \in \rho$. U^s_{tu} is the output of node u: thus, since for all $u' \in \rho\backslash\{v\}$, $u' \not\leq v$, then neither u nor any ancestor is in ρ. So, U^s_{tu} is unaffected by changing θ_ρ. More specifically, $U^s_{tu}(\mathbf{a}|_\rho(\tilde{\theta})) = U^s_{tu}(\mathbf{a})$. So:

$$\frac{\partial U^p(\mathbf{a}|_\rho(\tilde{\theta}))}{\partial \tilde{\theta}(u,v)} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{n}\sum_{\rho\in P(\rho,o_k)} U^s_{tu}(\mathbf{a}) q_{t,p_{|p|}} a_{kt} \prod_{j=1}^{|p|-1} \theta(p_j,p_{j+1}) q_{t,p_j} \tag{31}$$

So, the partial derivative is a function only of $\mathbf{a}$, not $\tilde{\theta}$. A function with a constant partial derivative along every coordinate is affine. ■

We now prove a more general version of Lemma 3.

Lemma 23 *Given a fixed protagonist action θ, there exists a unique joint action for all agents $\sigma = (\theta, \{a_t, b_t\}, \{q_{vt}, d_{vt}\})$ (the joint action expansion) where the zannis and the antagonist are playing best responses to σ. Moreover, $U^p(\sigma) = -L(\theta)$, $\nabla_\theta U^p(\sigma) = -\nabla L(\theta)$, and given some protagonist at $\rho \in P$, if we hold all other agents' strategies fixed, $U^p(\sigma)$ is an affine function of the strategy of the protagonist at p.*

Proof: Most of the insights are in Lemmas 18, 19, 20, and Lemma 22. We know from above that everyone will be reasonable in the joint action expansion. We just have to carefully construct it to prove that it exists and is unique. Consider a parameter $\theta \in \Theta$, and an arbitrary joint action $\mathbf{a}^0 = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$. First of all, given the partial ordering $\leq$ over V, consider $\sqsubseteq$ to be a linear extension of $\leq$, such that we construct $v_1 \ldots v_{|V|}$, where $v_k \sqsubseteq v_{k+1}$. Define $\mathbf{a}^k$ such that $\mathbf{a}^k$ is equal to $\mathbf{a}^{k-1}$, except that the zanni at v^k plays a best response to $\mathbf{a}^{k-1}$. Finally, $\mathbf{a}^*$ will be equal to $\mathbf{a}^{|V|}$ except that the adversary plays a best response. We prove recursively that each time a best response needs to be taken by a zanni, it exists and is reasonable, by Lemma 18. Thus, all the zannis are reasonable in $\mathbf{a}^{|V|}$, thus a best response for the antagonist exists and is reasonable in Lemma 19. Therefore, there does exist some joint action $\mathbf{a}^*$ when all zannis and the adversary are playing a best response. We can then prove that this is unique, again by using Lemma 18 and Lemma 19 guarantee that the reasonable action is a unique best response, and the reasonable action depends only upon θ.

Now, we have established that the joint action extension exists and is unique. We now want to prove the other properties described in Lemma 23.

Because the adversary is reasonable, by definition, for all t $a_t^\top c_t(\mathbf{o}, \theta) + b_t = l_t(c_t(\mathbf{o}, \theta))$ and $a_t = \nabla l_t(z)|_{z=c_t(\mathbf{o},\theta)}$. Because the zannis are reasonable, for all t, for all $o \in O$, $c_t(o, \theta) = U^s_{ot}(\mathbf{a})$.

Thus, by the definition of the utility of the antagonist:

$$U^a_t(\mathbf{a}) = b_t + \sum_{k=1}^{n} a_{kt} U^s_{o_k t}(\mathbf{a}) \tag{32}$$

$$U^a_t(\mathbf{a}) = b_t + \sum_{k=1}^{n} a_{kt} c_t(o, \theta) \tag{33}$$

$$U^a_t(\mathbf{a}) = l_t(c_t(\mathbf{o}, \theta)) \tag{34}$$

Therefore, averaging over t, $U^a(\mathbf{a}) = L(\theta)$. By Lemma 20, $\nabla_\theta U^{prot}(\mathbf{a}) = -\nabla_\theta L(\theta)$. Finally, from Lemma 22, every protagonist faces an affine utility function if she unilaterally deviates. ■

Proof (of Lemma 3): Lemma 3 is a special case of Lemma 23. ■

Lemma 24 *Assume that for all $v \in V$, f_v is convex and differentiable, the loss l is convex and partially differentiable in the first term, and given a joint action $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$ where all zannis are reasonable, and the adversary is reasonable, then if the joint action θ for the protagonists is a KKT point, then the protagonists actions are a best response to* $\mathbf{a}$*, and* $\mathbf{a}$ *is a Nash equilibrium.*

Proof: To prove that this is a Nash equilibrium, we need to show that for each $\rho \in P$, the protagonist at ρ is playing a best response (all zannis and adversaries are reasonable, so they are playing a best response). In other words, we need to show that, if we considered $U^p(\mathbf{a})$, as a function of the values of θ on $(u, v) \in E_\rho$, then the current θ in $\mathbf{a}$ is a global maximum. We do this in two steps.

1. We translate the KKT conditions for full problem with L to KKT conditions for a partial problem on $U^p_{\rho,\mathbf{a}}$, the utility function for the protagonist at v deviating.
2. Because $U^p_{\rho,\mathbf{a}}$ is affine, the KKT conditions for a maximum imply a global maximum (see Theorem 11).

As we established in Lemma 20:

$$\nabla_\theta U^p(\mathbf{a}) = -\nabla_\theta L(\theta) \tag{35}$$

Then, the KKT conditions on the loss imply that there exist KKT multipliers $\mu_{j,\rho}$ and $\lambda_{h,\rho}$ such that:

$$-\nabla L(\theta) = \sum_{\rho \in P} \sum_{j \in J_\rho} \mu_{j,\rho} \nabla j(\theta) + \sum_{\rho \in P} \sum_{h \in H_\rho} \lambda_{h,\rho} \nabla h(\theta) \tag{36}$$

$$\mu_{j,\rho} j(\theta) = 0 \text{ for all } \rho \in P, j \in J_\rho \tag{37}$$

Substituting equation 35:

$$\nabla_\theta U^p(\mathbf{a}) = \sum_{\rho \in P} \sum_{j \in J_\rho} \mu_{j,\rho} \nabla j(\theta) + \sum_{\rho \in P} \sum_{h \in H_\rho} \lambda_{h,\rho} \nabla h(\theta) \tag{38}$$

$$\mu_{j,\rho} j(\theta) = 0 \text{ for all } \rho \in P, j \in J_\rho \tag{39}$$

These are the necessary KKT conditions for θ to be a local maximum. But it is not sufficient. Choose a particular $\rho \in P$. Define $\theta_\rho \in \Theta_\rho$ to be the action of the protagonist at ρ in θ. Now, if we restrict this to the dimensions in E_ρ, only the constraints in J_ρ and H_ρ will vary, so:

$$\nabla_{\theta_\rho} U^p(\mathbf{a}) = \sum_{j \in J_\rho} \mu_{j,\rho} \nabla j(\theta) + \sum_{h \in H_\rho} \lambda_{h,\rho} \nabla h(\theta) \tag{40}$$

$$\mu_{j,\rho} j(\theta) = 0 \text{ for all } j \in J_\rho \tag{41}$$

We can replace $U^p(\mathbf{a})$ with $U^p_{\rho,\mathbf{a}}$. For the strategy θ_ρ that is a part of θ, we get:

$$\nabla_{\theta_\rho} U^p_{\rho,\mathbf{a}}(\theta_\rho) = \sum_{j \in J_\rho} \mu_{j,\rho} \nabla j(\theta_\rho) + \sum_{h \in H_\rho} \lambda_{h,\rho} \nabla h(\theta_\rho) \tag{42}$$

$$\mu_{j,\rho} j(\theta_\rho) = 0 \text{ for all } j \in J_\rho \tag{43}$$

These are the KKT conditions for θ_ρ to be a local maximum of $U^p_{\rho,\mathbf{a}}$ in Θ_ρ. Therefore, the protagonist at ρ cannot gain by deviating. Now, by Lemma 22, we know $U^p_{\rho,\mathbf{a}}$ is affine, and so if the KKT conditions for a local maximum are satisfied, so are the KKT conditions for a global maximum. Thus, this implies each protagonist cannot unilaterally[4] improve on $\mathbf{a}$, and therefore this is a Nash equilibrium. ■

Theorem 25 *Assume that for all $v \in V$, f_v is convex and differentiable, the loss l is convex and partially differentiable in the first term. For every KKT point $\theta \in \Theta$, there is a Nash equilibrium where the joint action of the protagonists is θ, and for every Nash equilibrium where the joint action of the protagonists is $\theta \in \Theta$, θ is a KKT point.*

Proof: To prove that the Nash equilibrium exists, consider the joint action extension of θ. This is a Nash equilibrium by Lemma 24.

To prove the converse, we run the argument of Lemma 24 in reverse. To prove that given a Nash equilibrium $\mathbf{a} = (\theta, \{a_t, b_t\}, \{q_{v,t}, d_{v,t}\})$, θ is a KKT point, first observe that for any Nash equilibrium, the zannis and adversaries are reasonable (because they are playing best responses). In other words, $\mathbf{a}$ is the joint action extension of θ. Therefore, $\nabla_\theta U^p(\mathbf{a}) = -\nabla_\theta L(\theta)$. Because the equilibrium is an optimal value for the affine function $U^p_{v,\mathbf{a}}$, the KKT conditions must hold for each protagonist. Combining the KKT conditions for each protagonist gives KKT conditions for maximizing U^p over Θ. Since $\nabla_\theta U^p(\mathbf{a}) = -\nabla_\theta L(\theta)$, we can translate the KKT conditions for maximizing U^p into the KKT conditions for minimizing $L(\theta)$. ■

Proof (of Theorem 4, DLG Nash Equilibrium): This is a variant of Theorem 25. ■

G Convexity of Antagonist's and Zanni's Strategy Space

This appendix is a side note and thus the notation is mostly disconnected from the rest of the paper. We do not claim that this is original, but it is important to understand whether antagonists can minimize regret.

In order to do online convex optimization, we must have a convex strategy space. Suppose you have a convex, differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. Consider the set C of all $a \in \mathbb{R}^n$, $b \in \mathbb{R}$ such that for all $z \in \mathbb{R}^n$, $a^\top z + b \leq f(z)$.

We want to prove C is convex. Due to various technical issues, it is harder than you think.

Lemma 26 *For any $a \in \mathbb{R}^n$, if there exists a $b \in R$ such that $(a, b) \in C$, then there exists a $v \in \mathbb{R}$ such that for all $c \leq v$, $(a, c) \in C$, and for all $c > v$, $(a, c) \notin C$.*

[4] Notice that in some cases multiple agents could improve on a KKT point. Thus we are proving that this is a Nash equilibrium, not a strong Nash equilibrium.

Proof: Since $(a, b) \in C$, for all z, $f(z) \geq a^\top z + b$. We can define a function $g(z) = f(z) - (a^\top z + b) \geq 0$ is bounded from below, and therefore has a greatest lower bound q. $v = b + q$. Thus, for all $z \in \mathbb{R}^n$, $g(z) \geq q$. Therefore, $f(z) \geq a^\top z + b + q = a^\top z + v$ for all $z \in \mathbb{R}^n$.

1. If $c \leq v$, then for all $z \in \mathbb{R}^n$, $f(z) \geq a^\top z + b + q \geq a^\top z + c$.
2. For any $c > v$, then $c - b > q$, and there exists a $z \in \mathbb{R}^n$ where $c - b > g(z)$. For this z, $f(z) < a^\top z + c$.

■

Lemma 27 *If* $(a^1, b^1) \in C$ *and* $(a^2, b^2) \in C$*, then for any* $\lambda \in [0, 1]$*, there exists some* $b^3 \in \mathbb{R}$ *such that* $(\lambda a^1 + (1 - \lambda)a^2, b^3) \in C$.

Proof: Define $g^1(z) = (a^1)^\top z + b^1$ and $g^2(z) = (a^2)^\top z + b^2$ Consider the function $g(z) = \max(g^1(z), g^2(z))$. By definition, for all $z \in \mathbb{R}^n$, $g(z) \leq f(z)$.

Now, there are three cases for $g(z)$:

1. for all $z \in \mathbb{R}^n$, $g(z) = g^1(z)$. If this is the case, then $a^1 = a^2 = \lambda a^1 + (1 - \lambda)a^2$, and therefore setting $b^3 = b^1$ works.
2. for all $z \in \mathbb{R}^n$, $g(z) = g^2(z)$. Same as above.
3. or there exists $z^1, z^2 \in \mathbb{R}^n$ such that $g(z^1) = g^1(z^1)$ and $g(z^2) = g^2(z^2)$. If this is the case, then there must exist a z^3 where $g(z^3) = g^1(z^3) = g^2(z^3)$. Since g is convex, and a^1 and a^2 are subgradients at z^3, then since the set of subgradients at a point is convex, $a^1\lambda + (1 - \lambda)a^2$ is a subgradient at z^3. Therefore, set $b^3 = g(z^3) - (a^1\lambda + a^2(1 - \lambda))^\top z^3$. Since the new function is below g, it is below f.

■

Theorem 28 *The set C described above is convex.*

Proof: Consider an arbitrary $(a^1, b^1) \in C$ and $(a^2, b^2) \in C$, and $\lambda \in [0, 1]$. For simplicity, define $a^3 = \lambda a^1 + (1 - \lambda)a^2$.

From Lemma 27, there exists a b^3 such that $(a^3, b^3) \in C$. Thus, from Lemma 26, there exists a v such that for all $c \leq v$, $(a^3, c) \in C$, and for all $c > v$, $(a^3, c) \notin C$. Thus, if $\lambda b^1 + (1 - \lambda)b^2 \leq v$, we have proven the theorem.

Let us prove this by contradiction: namely, assume $\lambda b^1 + (1 - \lambda)b^2 > v$. Define $\epsilon = \lambda b^1 + (1 - \lambda)b^2 - v$. Since by definition, $\sup_z((a^3)^\top z + v) - f(z) = 0$, then there must exist some $z \in \mathbb{R}^n$ such that $((a^3)^\top z + v) - f(z) \geq -\epsilon/2$.

Now, for this z, $a^1 z + b^1 \leq f(z)$, and so $(a^1)^\top z + b^1 \leq ((a^3)^\top z + v) + \epsilon/2$. Similarly, $(a^2)^\top z + b^2 \leq ((a^3)^\top z + v) + \epsilon/2$. Thus, we can combine these to show

$$(\lambda a^1 + (1 - \lambda)a^2)^\top z + \lambda b^1 + (1 - \lambda)b^2 \leq ((a^3)^\top z + v) + \epsilon/2 \tag{44}$$

By the definition of a^3:

$$(a^3)^\top z + \lambda b^1 + (1 - \lambda)b^2 \leq ((a^3)^\top z + v) + \epsilon/2 \tag{45}$$

$$\lambda b^1 + (1 - \lambda)b^2 \leq v + \epsilon/2 \tag{46}$$

However, we defined $\epsilon = \lambda b^1 + (1 - \lambda)b^2 - v$, so $v + \epsilon = \lambda b^1 + (1 - \lambda)b^2$, a contradiction. ■

H Convolutional Neural Networks

Convolutional networks are an extension of the simple feedforward neural network where edge parameters are tied in a particular manner. In particular, if we consider a convolutional layer $l \in L$ that has a width w_l, height h_l, and depth d_l, the vertices within the layer can be indexed by

$I_l = \{1 \dots w_l\} \times \{1 \dots h_l\} \times \{1 \dots d_l\}$; that is, an individual vertex in layer l can be denoted $v_{l,i,j,k}$. The convolution defined at layer l also has a window: for instance, a window of 5×5 for layer l means that there is an edge between $v_{l-1,i,j,k}$ and $v_{l,i',j',k'}$ if and only if $|i - i'| \leq 2$ and $|j - j'| \leq 2$. Moreover, two edges $(v_{l-1,i,j,k}, v_{l,i',j',k'})$ and $(v_{l-1,i'',j'',k''}, v_{l-2,i''',j''',k'''})$ in E have equal weight if and only if $i - i' = i'' - i'''$, $j - j' = j'' - j'''$, $k = k''$, and $k' = k'''$. All of these constraints are linear equalities and they all occur in the same layer; therefore we can partition the vertices by layer and obtain a valid partitioning, since no vertex is the ancestor of another vertex within the same layer. Moreover, the equality constraints are all valid for the local linear inequality constraint qualification. We can also add an L_1 or L_2 bound on the weights, either by the terminal vertex of the edge or for the entire layer. Notice that, in practice, instead of having equality constraints between edges, a single copy of the weights is sufficient.

I Non-Convex Activation Functions

At first glance, it might appear that the restriction to convex activation functions is too severe, in the sense that it does not include standard (differentiable) activations such as sigmoid and tanh. However, the ability to partition vertices and tie weights with linear equalities, as developed above, allows any activation function that can be expressed as a *difference* of convex functions to still be exactly modeled within our framework. For example, note that the sigmoid, $\sigma(z) = \frac{1}{1+e^{-z}}$, and $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ functions can each be written as a difference of differentiable convex functions: For the sigmoid we have $\sigma(z) = \sigma^+(z) - \sigma^-(z)$ for $\sigma^+(z) = \frac{1}{2}(z + \sigma(z) - \log \sigma(z))$ and $\sigma^-(z) = \frac{1}{2}(z - \sigma(z) - \log \sigma(z))$, which are both convex and differentiable. For tanh we have $\tanh(z) = \tau^+(z) - \tau^-(z)$ for $\tau^+(z) = 2\sigma^+(2z) - \frac{1}{2}$ and $\tau^-(z) = 2\sigma^-(2z) + \frac{1}{2}$ which are also both convex and differentiable.

In general, at a node v, we can consider any activation function f_v that can be written as a difference of functions, $f_v = f_v^+ - f_v^-$, such that f_v^+ and f_v^- are both smooth and convex. In such cases, we can then simulate the contribution of f_v to the circuit computation by adding two sub-nodes, v^+ and v^-, below v, connecting these to v via two new edges (v^+, v) and (v^-, v), and replacing each edge (u, v) by the pair of edges (u, v^+) and (u, v^-). Also, we assign the differentiable convex activation f_v^+ to v^+, the differentiable convex activation f_v^- to v^-, and replace f_v at v with the identity activation $\tilde{f}_v(z) = z$. Then to ensure f^+ and f^- receive the same input, we merely add the parameter tying constraints $\theta_{(u,v^+)} = \theta_{(u,v^-)}$ for each pair of corresponding incoming edges (u, v^+) and (u, v^-). To simulate the desired output value, we merely add the constraints that $\theta_{(v^+,v)} = 1$ and $\theta_{(v^-,u)} = -1$ for the parameters on the new edges (v^+, v) and (v^-, v). Denote the modified edge set by $\tilde{E}$. Then we have

$$\begin{aligned}
&\tilde{f}_v \left(\theta(v^+, v) f_{v^+}\Big(\sum_{u:(u,v^+)\in\tilde{E}} c_t(u,\theta)\theta(u,v^+) \Big) + \theta(v^-, v) f_{v^-}\Big(\sum_{u:(u,v^-)\in\tilde{E}} c_t(u,\theta)\theta(u,v^-) \Big) \right) \\
&\quad = f_{v^+}\Big(\sum_{u:(u,v^+)\in\tilde{E}} c_t(u,\theta)\theta(u,v^+) \Big) - f_{v^-}\Big(\sum_{u:(u,v^-)\in\tilde{E}} c_t(u,\theta)\theta(u,v^-) \Big) && (47) \\
&\quad = f_{v^+}\Big(\sum_{u:(u,v)\in E} c_t(u,\theta)\theta(u,v^+) \Big) - f_{v^-}\Big(\sum_{u:(u,v)\in E} c_t(u,\theta)\theta(u,v^+) \Big) && (48) \\
&\quad = f_v\Big(\sum_{u:(u,v)\in E} c_t(u,\theta)\theta(u,v^+) \Big). && (49)
\end{aligned}$$

Thus, $\theta(u, v^+)$ in the new graph is just like $\theta(u, v)$ in the old one. That is, the new circuit output at node v is the same as the original circuit output at node v, but now the neural network only uses differentiable convex activations at each vertex.

Another solution that works for any differentiable function is that, instead of trying to make the zanni maximize the output of the node, have zanni try to "guess" the derivative and the offset. Specifically, given z is the input to the activation function f_v for example t, define $q_{v,t}^* = f_v'(z)$, and $d_{v,t}^* = f_v(z) - f_v'(z)z$, and make the utility of the zanni at v to be $\sum_t (q_{v,t} - q_{v,t}^*)^2 + (d_{v,t} - d_{v,t}^*)^2$. As before, reasonable behavior for the zanni is the unique optimal behavior.

J Discussion of [2]

The unpublished manuscript [2] presents a variety of interesting and related ideas. In the "gated game" and "CoG game" proposed therein, agents similar to our protagonists are introduced at every vertex. In the gated game, gates are introduced that act as a function of the strategies of the protagonists. In the CoG game, agents (like zannis) are introduced whose actions are a function of the protagonists' actions.

However, the game representations proposed here and in [2] are fundamentally different. The utilities and available information to an agent in a game is crucial in game theory. For instance, Stackelberg games and simultaneous move games are fundamentally different. In a Stackelberg game, one player moves first, and the second observes their movement. In a simultaneous move game, both players must select their strategy independently. For instance, consider a cooperative game, where two players each get a dollar if they both say "heads" or both say "tails", but nothing if they say something different. If one assumes both players move at the same time versus one after the other, these are very different games. Moreover, there is a difference between an agent that is motivated to take an action, versus one that is restricted to play a certain action.

The reason that these distinctions are important is that conventional approaches of regret minimization work in the game developed in this paper but not in [2]: here there is no need to define a new type of regret that is particular to deep networks. Given that a key contribution of this paper is a way to think about optimization problems, whether the concept corresponds to a conventional notion of regret or a new notion of regret is quite important. We leverage the technique in [17] where a game is played with one agent minimizing regret and the other playing a best response. Such a result is not magical: in game theory, there is a huge distinction between having a perfect model of your opponent and prescient knowledge of their actions: the former is still a simultaneous move game, whereas the latter is a Stackelberg game with different equilibria. If the protagonists use randomness, then their behavior cannot be predicted perfectly, and [17] cannot be applied. It is also key that zannis and adversaries observe the example selected by chance: otherwise, they would not be able to model the utilities experienced by the protagonist.

A further distinction is the quality of the connection between solution concepts in the learning problem and the proposed game. in this paper, we show that there is a bijection between Nash equilibria and "KKT points". KKT points include both true local minima as well as some saddle points. Thus, this is a more thorough understanding than the one-way implication in [2] about potential games, that local minima of the potential function are pure Nash equilibria. Since we are viewing the game as a window into the minimization, not being able to account for all Nash equilibria is a limitation. Moreover, it is hard to rectify this issue in [2], since there is no standard equivalent to KKT points for non-differentiable, non-convex functions.

An exact potential game is a game where the utilities are equal. [2] makes references to potential games: "Moreover, simple algorithms such as fictitious play and regret-matching converge to Nash equilibria in potential games," yet the given references have results for two player potential games with a finite number of actions [16, 14]. For the gated game, it is unclear whether these results would extend, especially given the complex nature of the loss functions introduced. It is an open question whether regret minimizers in multiplayer (i.e., more than two) potential games converge to a Nash equilibrium, and whether those results hold for more complex strategy spaces. Moreover, for the purposes of the games in this paper, it is important to understand if convergence results hold for games where only a subset of the agents have utilities that are equal, but one can make strong statements about the behaviors of the other agents.

In this paper, we have focused on deep networks with differentiable activation functions and losses. To deal with issues in non-differentiable activation functions, [2] introduces gated games, but these mean that the games are not in a standard form. The gates are sometimes considered to be dependent upon the agent's behaviors (as in when the game is considered as a potential game) and sometimes the gates are considered to be independent of the agent's behaviors (as when minimizing regret); this is partially justified by GRegret, but not the nuances introduced. There are unsupported statements, such as that minimizing regret guarantees correlated equilibrium (it is actually minimizing internal regret guarantees correlated equilibria, not external regret, and GRegret is clearly more closely associated with GRegret). Thus, one cannot use convergence results about games with a finite number of actions without extending said results to the case of gated games.