NEURON AS AN AGENT

Anonymous

ABSTRACT

1 INTRODUCTION

Deep reifnorce ment learning (DRL) succeed in many area. Deep Q-Network (DQN) (Mnih et al., 2015; Silver et al., 2016) descides the optimal action from screen sequence of atar, and selects the move closest to win from a face of a board of Go. Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015) realizes the multiple-join control considering condition such as friction and gravity factor in a physical space. The applicable are of DRL is becoming wider year by year, the reasonable performance is reported 3D game such as Doom (Dosovitskiy & Koltun, 2016).

The reason why a neural network is workable for DRL is that a neural network abstracts the implicit state in an environment, and obtains informative state representation. From the micro perspective, the abstraction capability of each unit contributes to return of the entire system. So, we address the one following question.

Will reinforcement learning work even if we consider each of units as an autonomous agent?

The contribution of this paper is that, we propose *Neuron as an Agent* (NaaA) as a novel framework for RL, and show its optimizing mehod. NaaA considers all the units in a neural network as agents, and optimizes the reward distribution as a multi-agent RL problem. In the reward design of NaaA, a unit distributes its received reward to other input units passing its activation to the unit as cost. Hence, the actual reward is profit which defined as difference between inflow (received reward) and outflow (paid cost). In the setting, the economic metaphor can be introduced: profit is balance of revenue and cost. It means that a unit should address trade-off between both optimization of cumulative revenue maximization and cumulative cost minimization.

This paper is organized as below. Firstly, with showing the optimization of NaaA, we report the negative result that the performance decreases if we naively consider the units as agents. As solution of the problem, we introduce a mechanism of auction which applying game theory. As theoretical result, we show that the agent obeys to maximize its *counterfactual return* as the Nash equilibrium. Counterfactual return is the one which we extend counterfactual reward, the criterion which proposed for multi-agent reward distribution problem (Agogino & Tumer, 2006), along time axis.

After that, we show that learning counterfactural return leads the model to learning optimal topology between the units, and propose *adaptive dropconnect*, a natural extension of dropconnect (Wan et al., 2013). Adaptive dropconnect combines dropconnect, which pure-randomly masks the topology, with adaptive algorithm, which prunes the connection with less counterfactual return with higher probability. It uses ε-greedy as a policy, and is equivalent to dropconnect in the case of $\varepsilon = 0$, and is equivalent to counterfactual return maximization which constructs the topology deterministically in the case of $\varepsilon = 1$.

At the last, we confirm that optimization with the framework of NaaA leads better performance of RL, with numerical experiments. Specifically, we use a single-agent environment from Open AI gym, and multi-agent environment from ViZDoom.

Although considering all the units as agents might be vacuity at first glance, it has wider applicable area. At the perspective of optimization for single neural network, it can apply to pruning by optimizing the topology. Not only that, introducing the concept of reward distribution divides the single neural network to a lot of autonomous parts. It enable us to not only address sensor placing problem in IoT for partially observed Markov decision process (POMDP), but arbitrary incentivized participants can join the framework.

2 RELATED WORK

Training neural network with multi-agent game is emerging methodology. Generative adversarial nets (GAN) (Goodfellow et al., 2014) has goal to obtain true generative distribution as Nash equilibrium of a competitive game made of two agents with contradicting rewards: a generator and a discriminator. In game theory, the outcome maximizing overall reward is named Pareto optimality. Nash equilibrium is not guaranteed to converge Pareto optimality, and difference of the both is named dilemma. As existence of dilemma depends on the reward design, the method to resolve the dilemma with good reward design is being researched: mechanism design (Myerson, 1983) also known as inverse game theory. Mechanism design is applied to auction (Vickrey, 1961) and matching (Gale & Shapley, 1962). GAN and our proposal, NaaA, are outcome from mechanism design. NaaA applies digital goods auction (Guruswami et al., 2005) to reinforcement learning with multi-agent neural network, and obtain maximized return by units as Nash equilibrium.

NaaA belongs to a class of partially observable stochastic game (POSG) (Hansen et al., 2004) as it processes multiple units as agents. POSG is a class of reinforcement learning in which multiple agents in a POMDP environment, and it has several research issues. The one is communication. CommNet (Sukhbaatar et al., 2016) exploits the characteristics of a unit which agnostic to topology of other units, it employs backpropagation to training multi-agent communication. The another one is credit assignment. Instead of reward $R(a_t)$ of an agent i for actions at t a_t, QUICR-learning (Agogino & Tumer, 2006) maximizes counterfactual reward $R(a_t) - R(a_t - a_{it})$, the difference in the case of the agent i takes an action a_{it} (a_t) and not $(a_t - a_{it})$. COMA (Foerster et al., 2017) also maximizes counterfactual reward in a setting of actor-critic. In the setting, all the actors has common critic, and improves the both actors and critic with time difference (TD)-error of counterfactual reward. This paper unifies both the issues, communication and credit assignment. The main proposal is framework to manage the agents to maximize *counterfactual return*, the extended counterfactual reward along with time axis.

TODO: Dropconnect

3 BACKGROUND

First, we consider a POMDP environment in which a single agent acts. POMDP environment is a 7-tuple $(\mathcal{S}_{\mathrm{H}}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{S}_{\mathrm{O}}, \mathcal{O}, \gamma)$, where $\mathcal{S}_{\mathrm{H}}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{T}$ is a transitive probability, $\mathcal{S}_{\mathrm{O}}$ is a possible set of observation, $\mathcal{O}$ is a set of observation probability, and γ is discount rate. An agent predicts partially state $h \in \mathcal{S}_{\mathrm{H}}$ through an observation $s \in \mathcal{S}_{\mathrm{O}}$. Generally, s has higher dimension than h, and is complex. For example, although Atari 2600 has a read only memory (RAM) as the true state, which contains 128 bytes, the generated image from that s has more than 10,000 dimension. Hence, DQN and DRQN abstracts s, and creates original state representation to predict good action efficiently. (although the original paper of DQN assumes MDP, the paper of DRQN pointed out that the environment is POMDP). Though DQN does not address the state transition directly because it is model-free method, some interpretation holds that the hidden state representation is learned in the previous layer of the output layer (Zahavy et al., 2016) In the method below, we assume that the agent decides the action through a neural network.

TODO: POSG

The design of NaaA is inspired by neuroscience. A neuron in a neurocircuit consumes adenosine triphosphate (ATP) supplied from connected astrocytes. The astrocyte is a glia cell, which forms structure of a brain, and it supplies fuel from vessel. As the amount of ATP is limited, the discarded neuron will become extinct with executing apoptosis. As aspotosis of a neuron is restrained by neurotorophin (NTF) such as nerve growth factor (NGF) and brain-derived neurotrophic factor (BDNF), The neuron which can obtain a lot of NTF will live. The perspective to interpret a neuron as an independent living object is kwown as neural Darwinism (Edelman, 1987).

4 NEURON AS AN AGENT

TODO: Show the figure.

A typical artificial neural network is a directed graph $\mathfrak{G} = (\mathcal{V}, \mathcal{E})$ among the units. $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a set of the units, and $\mathcal{E} \subset \mathcal{V}^2$ is a set of edge indicating connection between two units. If $(v_i, v_j) \in \mathcal{E}$, then connection $v_i \to v_j$ holds, indicating v_j observes activation of v_i. We denote activation of the unit v_i at time t as $x_{it} \in \mathbb{R}$. Also, we denote a set of units which unit i connects to as $N_i^{\text{out}} = \{j|(v_i, v_j) \in \mathcal{E}\}$, and a set of units which unit i is connected from is $N_i^{\text{in}} = \{j|(v_j, v_i) \in \mathcal{E}\}$. We denote $N_i = N_i^{\text{in}} \cup N_i^{\text{out}}$.

NaaA interprets v_i as an agent. Hence, $\mathfrak{G}$ is a multi-agent system. An environment for v_i is made of an environment which the multi-agent system itself touches to, and set of the unit which v_i directly connects to: $\{v_i \in V|i \in N_i\}$. We distinguish the both environments by naming the former as an external environment, and latter as an internal environment. v_i will receive reward from both the environments. We add the following assumption as the characteristics of v_i.

- N1: (Selfishness) Instead of minimizing the global training error, at each timing t, v_i acts to maximize toward maximizing its own return (cumulative discounted reward) $G_{it} = \sum_{k=0}^{T} \gamma^k R_{i,t+k}$, where $\gamma \in [0, 1]$ is discount rate, T is terminal time.
- N2: (Conversation) The summation of reward which $\mathcal{V}$ will receive both internal and external environment R_{it} over all the units equivalents to reward R_t^{ex} which the entire multi-agent system receives from the external environment.
- N3: (Trade) v_i receives internal reward ρ_{jit} from $v_j \in \mathcal{V}$ in exchange of activation signal x_i before transferring the signal to the unit. At the same time, ρ_{jit} is subtracted from the reward of v_j.
- N4: (NOOP) v_i has NOOP (no operation) in which the return is $\delta > 0$ as an action. With NOOP, the unit inputs nothing, and outputs nothing.

In terms of neuroscience, N1 states that the unit act as a cell. N2 and N3 state distribution of NTF, and N4 corresponds to apoptosis. NOOP is selected when expected return of the other actions are non-positive. In the following, we construct the framework of NaaA getting off from the assumptions.

5 NEURON AS AN AGENT

ニューラルネットワークを、ユニット間の有向グラフ $\mathfrak{G} = (\mathcal{V}, \mathcal{E})$ で表す。$\mathcal{V} = \{v_1, \ldots, v_N\}$ はユニットの集合であり、$\mathcal{E} \subset \mathcal{V}^2$ はユニットの接続関係を表すエッジの集合である。$(v_i, v_j) \in \mathcal{E}$ であるとき、$v_i \to v_j$ という接続関係が成立し、v_j は v_i から値を入力する。ユニットの v_i の時刻 t における出力を $x_{it} \in \mathbb{R}$ で表す。ユニット i の出力先の集合を $N_i^{\text{out}} = \{j|(v_i, v_j) \in \mathcal{E}\}$、入力元の集合を $N_i^{\text{in}} = \{j|(v_j, v_i) \in \mathcal{E}\}$ で表現する。$N_i = N_i^{\text{in}} \cup N_i^{\text{out}}$ とする。

NaaA は v_i をエージェントとしてとらえる。すなわち、$\mathfrak{G}$ はマルチエージェントシステムである。v_i にとっての環境は、マルチエージェントシステムの自体が触れている環境と、v_i が直絶接続しているユニット群 $\{v_i \in V|i \in N_i\}$ である。前者を外的環境 (external environment)、後者を内的環境 (internal environment) と呼んで区別する。v_i は環境から報酬を受け取る。v_i の性質として以下の前提を加える。

- N1: （利己性）v_i は、各時点 t において、汎化誤差の最小化ではなく、自身のリターン（累積減衰報酬）$G_{it} = \sum_{k=0}^{T} \gamma^k R_{i,t+k}$ の最大化を目的として行動する。ただし $\gamma \in [0, 1]$ は減衰率 (discount rate)、T は終端時間である。
- N2: （保存則）v_i が内的環境と外的環境の両方から受け取る報酬 R_{it} の総和は、マルチユニットシステム全体が外的環境から得る報酬 R_t^{ex} に等しい。
- N3: （取引）v_i は信号 x_i を $v_j \in \mathcal{V}$ に伝達する際に、信号と引き換えに報酬 ρ_{jit} を受け取る。同時に ρ_{jit} は v_j の報酬から差し引かれる。
- N4: （NOOP）v_i は、期待リターンが $\delta > 0$ の NOOP (no operation) という行動をオプションとして持つ。NOOP では、ユニットは何も入力せず、何も出力しない。

N1 はユニットがエージェントとして振る舞うことを述べている。N2, N3 は NTF の分配に、N4 はニューロンのアポトーシスに相当する。NOOP が選択されるのは、それ以外のすべての行動の期待報酬が非正であった場合である。以下ではこれらの前提から出発して、NaaA の仕組みを構築していく。

5.1 CUMULATIVE DISCOUNTED PROFIT MAXIMIZATION FRAMEWORK

ユニット i が時刻 t で外的環境から得る報酬を R_{it}^{ex} と書く。ただし、$\sum_{i=1}^{n} R_{it}^{\mathrm{ex}} = R_t^{\mathrm{ex}}$ である。N3 より、時刻 t に i が獲得する報酬 R_{it} は次のように表現される。

$$R_{it} = R_{it}^{\mathrm{ex}} + \sum_{j \in N_i^{\mathrm{out}}} \rho_{jit} - \sum_{j \in N_i^{\mathrm{in}}} \rho_{ijt}. \tag{1}$$

この式は、符号が正の項と負の項の二つに分解される。前者を収益 (revenue)、後者をコスト (cost) と呼び、それぞれ $r_{it} = R_{it}^{\mathrm{ex}} + \sum_{j \in N_i^{\mathrm{out}}} \rho_{jit}$, $c_{it} = \sum_{j \in N_i^{\mathrm{in}}} \rho_{ijt}$ で表す。R_{it} を利益 (profit) と呼ぶ。

このとき、ユニット v_i は、次式で表現される累積減衰利益 G_{it} を最大化する。

$$G_{it} = \sum_{k=0}^{T} \gamma^k R_{i,t+k} = \sum_{k=0}^{T} \gamma^k (r_{i,t+k} - c_{i,t+k}) = r_t - c_t + \gamma G_{i,t+1} \tag{2}$$

ここで、G_{it} はエピソードの最後になるまで明らかにならない。最適な行動を選択するためには、現在までの値に基づいた予測を行う必要があるため、G_{it} を価値関数 (value function) $V_i^{\pi_i}(s_{it}) = \mathbb{E}_{\pi_i}\left[G_{it} \mid s_{it}\right]$ で近似する。この時、次式が成立する。

$$V_i^{\pi_i}(s_{it}) = r_{it} - c_{it} + \gamma V_i^{\pi_i}(s_{i,t+1}), \tag{3}$$

このため、即時収益、価値関数の最大化と、即時コストの最小化についてそれぞれ考えればよい。$R_{it} > 0$、すなわち、$r_{it} > c_{it}$ であれば、ユニットは得たデータに対して付加価値を与えていることになる。もし、すべての t に対して $R_{it} \leq 0$ であれば、$V_i^{\pi_i}(s_{it}) \leq 0 < \delta$ であるから、ユニットは NOOP になる。

6 OPTIMIZATION

NaaA では利益を最大化するため、二つの相反する指標である収益 r_{it} とコスト c_{it} のバランスを取ることが重要になる。本研究では、この最適化にゲーム理論の一つであるメカニズムデザインを応用する。メカニズムデザインは、マルチエージェントシステムを対象にしたゲーム理論の分野であり、各エージェントが利己的であることを想定した上で、システム全体が最適になるような帰結を目指すメカニズムの設計を目的とするものである。

TODO: 否定的な結論

メカニズムデザインを導入する理由は、NaaA いくつかの既存研究と異なり、すべてのエージェントが協力的ではなく、利己的であると仮定していることに起因している。前述の問題は、そのまま最適化すると報酬額は 0 に収束するため、すべてのニューロンが NOOP になるという trivial な解が得られる。マルチエージェントシステムは無情報でアクションを選択する必要が生じ、これはランダムなアクションを取っている状況に等しい。したがって、明らかに外的環境からの報酬 R_t は小さくなる。

このように最適化を行った結果、システム全体が最適化されない現象はジレンマとして知られており、囚人のジレンマ問題をはじめとし様々な研究が行われている。しかし、一般にエージェントが利己的であると仮定した場合の最適化は難しいとされている。メカニズムデザインはこうした問題を解決することができる。

6.1 ENVY-FREE AUCTION

パレート効率な仕組みを作るために、我々はオークション理論における digital goods auction からアイデアを借りる。オークション理論は、ゲーム理論におけるメカニズムデザインという分野に属しており、複数のエージェントの利害を一致させ、全体としてパレート最適を目指すことを目指している。digital goods auction は、本や音楽などの、複製可能な財を割り当てる仕組みを作っている。

Digital goods auction にはいくつかバリエーションがあるが、本研究では単純な前提のみを設けるだけでよいという理由から envy-free auction (Guruswami et al., 2005) を用いる。これは、同じ時点取引において、一つのユニットの価格を同じにするというものである。NaaA において、これは次の前提によって表現できる。

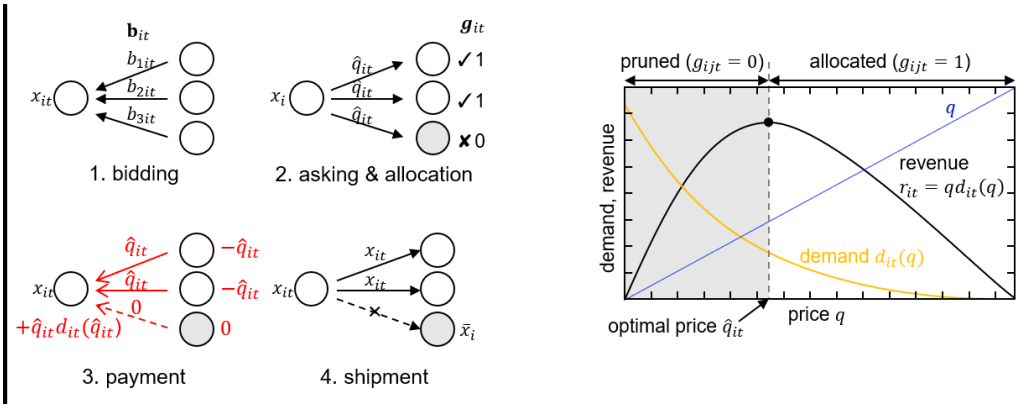

Figure 1: Left: NaaA による取引の流れ。Right: ユニットの価格決定方法。ユニットの収益は単調減少な需要と価格の積となり、これを最大化する価格が最適価格となる。

N5: （一物一価）$\rho_{j_1,i,t}, \rho_{j_2,i,t} > 0$ であれば $\rho_{j_1,i,t} = \rho_{j_2,i,t}$

これは、ユニット v_i は同じ時間 (timing) t に個有の価格を持つことを意味する。この価格を q_{it} で表す。

Envy-free auction の流れを Figure 1 の左に示す。図は、信号を送信する一つのユニットと、その信号を「購入」する複数のユニットに分かれ、交渉の過程を示している。一単位の交渉は、強化学習の時間軸では 1 ステップ内に完了し、これが複数回繰り返されることになる。信号を送信する側を売り手、受信する側を買い手と呼ぶ。買い手はユニットに対して入札 b_{jit} を行う (1)。次に、入札額をもとに、売り手は価格 $\hat{q}_{it}$ を決定し、割当を行う (2)。このとき、$b_{jit} \geq \hat{q}_{it}$ であれば割当を行って $g_{jit} = 1$ とし、そうでなければ $g_{jit} = 0$ とする。割当を行った後は、$\rho_{jit} = g_{jit}\hat{q}_{it}$ として、送金を行い (3)、売り手は割当を行ったノードに対してのみ信号 x_i を送付する (4)。信号を受け取れなかったノードは、x_i の期待値 $\mathbb{E}_\pi[x_i]$ によって x_i を近似する。

以下では、式 3 に基づき、収益、コスト、価値関数についてそれぞれ述べる。

Revenue: エージェントの収益は次式で与えられる。

$$r_{it} = \sum_{j \in N_i^{\text{out}}} g(b_{jit}, q_{it}) q_{it} + R_i^{\text{ex}} = q_{it} \sum_{j \in N_i^{\text{out}}} g(b_{jit}, q_{it}) + R_i^{\text{ex}}$$
$$= q_i d_{it}(q_t) + R_i^{\text{ex}}, \tag{4}$$

ただし、$g(\cdot,\cdot)$ は割当 (allocation) であり、ステップ関数 $H(\cdot)$ を用いて $g(b,q) = H(b-q)$ によって定義される。q_{it} は価格、$d_{it}(q_{it})$ はユニット i の信号に対する価値を q_{it} 以上と評価しているエージェントの数であり、需要 (demand) と呼ぶ。同様の式で、右辺を最大化する a を最適価格と呼び、$\hat{q}_{it}$ で表す。第二項は q_t に対して独立であるから、最適価格 $\hat{q}_{it}$ は次のようにして与えられる。

$$\hat{q}_{it} = \underset{q \in [0,\infty)}{\operatorname{argmax}}\, q d_{it}(q). \tag{5}$$

この仕組みを、Figure 1 の右に図示する。$d_t(q_{it})$ は単調減少な関数であり、収益 r_{it} は q_{it} と $d_t(q_{it})$ との積によって表現される。

Cost: コストは、ユニットが他のユニットに対して払う価格である。これは次のように表示される。

$$c_{it} = \sum_{j \in N^{\text{in}}} g(b_{ijt}, q_j) q_j \tag{6}$$

c_{it} 自体は $b_{ijt} = 0$ のとき最小となる。しかし、これは次の value function とトレードオフをなす。

Value Function: 価値関数は $V(s_{i,t+1})$ の値は $s_{i,t+1}$ に依存する。既に述べたようにエージェントの v_i の環境は接続されているユニット集合であり、ユニットの出力はこれらのエージェン

トからの評価、すなわちエッジの重みに影響を及ぼす。通常のニューラルネットワークでは、出力の精度に貢献しないニューロンの重みは小さくなることから、報酬は小さくなる。したがって、入札価格 b_{ijt} を最小化し 0 と置くとデータの購入に失敗し、将来的にエージェントが接続しているエージェントから得られる報酬が小さくなる。

今、割当を $\mathbf{g}_{it} = (g_{i1t}, \dots, g_{iNt})^{\mathrm{T}}$ で表し、エージェントが v_j の購入に成功した場合と、そうでない場合に価値関数に及ぼす影響について考える。この時、価値関数は、状態価値関数 $Q(s_{i,t+1}, \mathbf{g}_{i,t+1})$ を用いて次式で表現できる。

$$\begin{aligned} V_i^{\pi_i}(s_{it}) &= Q_i^{\pi_i}(s_{it}, \mathbf{g}_{it}) \\ &= \sum_{j \in N_i^{\mathrm{in}}} g_{ijt}(Q_i^{\pi_i}(s_{it}, \mathbf{e}_j) - Q_i^{\pi_i}(s_{it}, \mathbf{0})) + Q_i^{\pi_i}(s_{it}, \mathbf{0}) \\ &= \sum_{j \in N_i^{\mathrm{in}}} g_{ijt} o_{ijt} + Q_i^{\pi_i}(s_{it}, \mathbf{0}) \\ &= \mathbf{g}_{it}^{\mathrm{T}} \mathbf{o}_{it} + Q_i^{\pi_i}(s_{it}, \mathbf{0}) \end{aligned} \tag{7}$$

$o_{ijt} = Q_i^{\pi_i}(s_{it}, \mathbf{e}_j) - Q_i^{\pi_i}(s_{it}, \mathbf{0})$ を *counterfactual return* と呼ぶ。これは、counterfactual reward の cummulative discounted summation である QUICR (Agogino & Tumer, 2006) と導出は異なるが等価である。すなわち、エージェントが支払うコストは、データの購入に成功した場合は $\hat{q}_{it}$ であり、それ以外は o_{it} となる。

以上から、最適化問題は次のように書くことができる。

$$\max_{\mathbf{b}, q} V_i^{\pi_i}(s_{it}) = \max_{q} q d_{it}(q) - \min_{\mathbf{b}} \mathbf{g}_{it}(\mathbf{b})^{\mathrm{T}}(\hat{\mathbf{q}}_t - \gamma \mathbf{o}_{i,t+1}) + \mathrm{const.} \tag{8}$$

では、リターンを最大化するためのエージェントの入札額 b_{it} は何か。これについては次の定理が成立する。

Theorem 6.1. *(Truthfulness)* リターンを最大化する最適な入札額は $\hat{\mathbf{b}}_{it} = \mathbf{o}_{it}$ である。

証明については Appendix を参照。

すなわち、エージェントは自身の counterfactual return のみを問題にすればよい (!) したがって、NaaA のメカニズムでは、エージェントはあたかも他のエージェントを価値評価 (valuation) し、その価値を正直に申告していることを意味する。

系として次の解が得られる。

Corollary 6.1. *The Nash equivalem of the envy-free game* $(\mathbf{b}_{it}, q_{it})$ *is* $(\mathbf{o}_{it}, \operatorname{argmax}_q q d_{it}(q))$.

6.2 VALUATION NET

残る問題は、$\mathbf{o}_t$ をいかに推定するかである。この推定には様々な方法が存在しており、多くのメソッドを使うことができるが、本論文では Q の推定に Q-learning を採用する。ただし、SARSA や actor-critic などの on-policy な方法も使うことができることを補足する。

図 2 に示す Valuation Net は、通常のニューラルネットワークのユニットに、Q-learning による valuation を組み合わせたネットワークである。まず、上部はエージェント間の通信について示したものである。ニューラルネットワークではユニットを円で表現するのが通例であるが、ここではユニットをエージェントとしてみなすことを強調して、六角形で一つのユニットを表現している。エージェント間では、通常のニューラルネットワークと同様の信号の通信以外に、取引に関する通信 (allocate, buy, sell & bid) が発生する。

Valuation Net では、状態 $\mathbf{s}_t$ として、予測後入力 $\tilde{\mathbf{x}}_t$ および入力に依存しない構成情報 $\mathbf{u}$ を横につなげたベクトル $(\tilde{\mathbf{x}}_t^{\mathrm{T}}, \mathbf{u}^{\mathrm{T}})^{\mathrm{T}}$ を用いる。構成情報の一例としてはユニットのパラメータがあげられ、たとえば重みやバイアスの情報を用いることができる。

状態からの Q 関数の予測にニューラルネットワークを用いる。エージェントが受け取った収益に基づき時間差分 (TD)-誤差 が計算され、ネットワークが訓練される。ネットワークの構成にはこれまでの deep Q-learning で用いられている二重化ネットワーク (dueling network) (Wang et al., 2015) のテクニックを用いる。オリジナルの文献 (Wang et al., 2015) で述べられている二

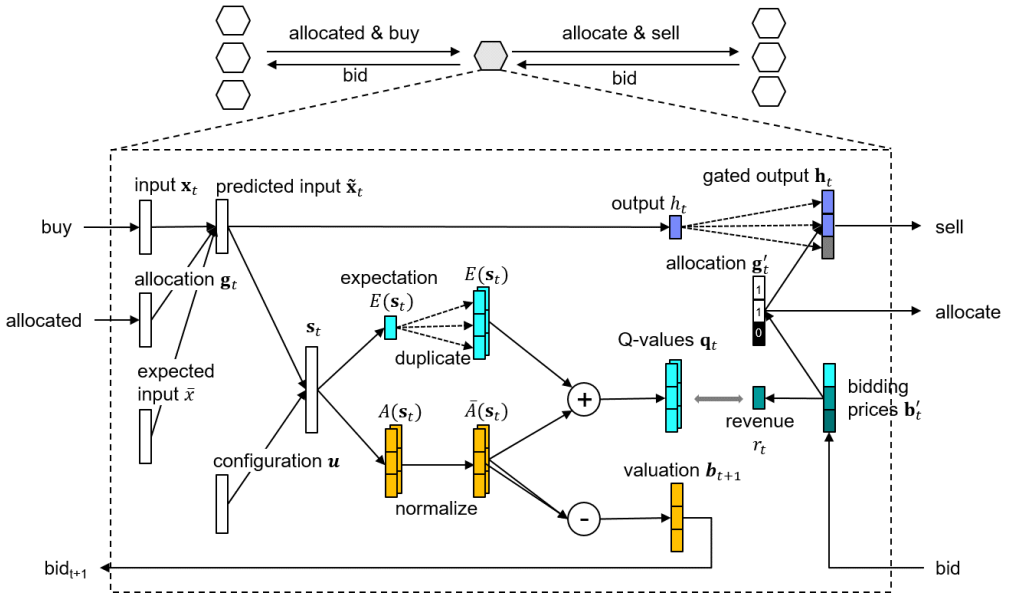

Figure 2: Valuationn Net は情報の価値を評価し、bidding price を決定する。下位のニューロンに対して入札し、信号を購入する。購入したデータを用いて、データを次のニューロンに対して売る。

重化ネットワークは、学習を加速するために、状態関数と、Q 関数との差分を別々に予測する手法である。Dosovitskiy & Koltun (2016) はこれに対して、差分の要素の総和が 0 になるように正規化するよう改良している。本研究では Dosovitskiy & Koltun (2016) の手法に従い、期待値 $\mathcal{E}(\mathbf{s}_t)$ と正規化差分 $\tilde{A}(\mathbf{s}_t)$ を別々に求める。

Q 関数は次のように表現される。

$$Q(\mathbf{s}_t, a_t) = \mathcal{E}(\mathbf{s}_t) + \tilde{A}(\mathbf{s}_t, a_t)$$
$$\sum_{i+1}^{k} \tilde{A}_i(\mathbf{s}_t, a_t) = 0 \tag{9}$$

第 2 式を満たすために、まず、$\mathbf{s}_t$ に基づいた予測を行い、次のような正規化を行う。

$$\tilde{A}_i(\mathbf{s}_t, a_t) = A_i(\mathbf{s}_t, a_t) - \frac{1}{k}\sum_{j=1}^{k} A_j(\mathbf{s}_t, a_t) \tag{10}$$

次に、valuation を行い、bidding price $\mathbf{b}_t$ を求める。$\hat{b}_{ijt} = o_{ijt}$ 式 9 より、最適な入札価格 $\hat{b}_{it}$ は次のように計算できる。

$$\hat{b}_{ijt} = \tilde{A}(\mathbf{s}_t, 1) - \tilde{A}(\mathbf{s}_t, 0) \tag{11}$$

Valuation Net ではこの式に基づき、advantage の出力を引き算することで入札価格を計算している。

7 Experiment

7.1 Single-agent environment

7.2 Multi-agent environment

7.2.1 ViZDoom

To 阿久澤君: ここの執筆をお願いできないでしょうか

8 DISCUSSION

8.1 DISADVANTAGE

Disdvantage としてまず挙げられるのは計算量である。Envy-free auction では需要の計算にソートの演算が入るために、直列化しなければならない箇所があるため、これらについては近似を行うなどして改善していく必要がある。

個別の最適化技術について述べると、Envy-free auction は、買い手のエージェント同士の価格がわからない sealed な状態であれば、正直性 (truthfulness) が成り立つが、一方で買い手同士がコミュニケーションを行い価格を共有し合う状態においては、買い手が自由に価格を偽装できることが知られている。これについては、Goldberg et al. (2006) によって解決方法が示唆されている。

Valuation Net は、用いるニューラルネットワークによっては実装が困難であることがある。これは著者らの GitHub に Linear と CNN は公開しているが、RNN などについては今後の研究課題となる。

8.2 APPLICATION

NaaA は、ネットワークが分散されている環境での学習や、サブモジュールでの制御に有用である。具体的に、以下の技術に応用が可能である。

- ハイパーパラメータチューニング。Neuroevolution など、遺伝的アルゴリズムを用いてハイパーパラメータチューニングを用いるアルゴリズムがすでにいくつか提案されている。このとき、fitness 関数として利益を用いることで、より強化学習の目的に特化したニューラルネットワークを得ることができると考えられる。
- pruning, dilusion などのネットワークの規模の縮小。
- アテンション制御。一部のアテンションの研究では、強化学習を用いてアテンションの制御を行っている。
- アンサンブル。複数のモデルの混合に今回の技術を用いることができる。

これらの応用に関しては、今後の研究の方向性である。

9 CONCLUSION AND FUTURE WORKS

本論文では、POMDP の問題設定において良質な特徴表現を得るために、ニューラルネットワーク上の各ユニットをエージェントとして扱うフレームワーク、NaaA について述べた。NaaA のフレームワークでは、ジレンマ問題を解決し、それぞれのエージェントの持つ付加価値がナッシュ均衡として得られ、全体としてパレート最適になることを示した。入札価格の決定アルゴリズムの一つとして、Q-learning に基づくネットワーク Valuation Net を示した。評価実験では、Atari と VizDoom を用いた実験を行い、実験結果が既存手法よりもよくなることを示した。

今後の方向性として、高速化、Valuation Net を A3C などの on-policy な手法で置き換えるといった方向性の他、神経科学的な説明を可能にしていくといった方法、遺伝的アルゴリズムとの組み合わせが挙げられる。

APPENDIX

A.1 定理 5.1 の証明

買い手の獲得する生涯報酬 G は次で与えられる。

$$G(b, q) = g(b, q) \cdot (v - q) + G_0, \tag{12}$$

ただし、g は割当 (allocation) であり、G_0 はユニットを購入しなかった場合の生涯報酬である。割当は、$g(b, q) = H(b - q)$ が成立する。H はステップ関数である。

買い手にとって売り手が提示する asking price q は未知であるため、q を台 $[0, \infty)$ の上の確率変数であるとして扱い、期待値 $\mathbb{E}_q[G(b,q)]$ を最大化することを考える。このとき、

$$\begin{aligned}\frac{\partial}{\partial b}\mathbb{E}_q[G(b,q)] &= \frac{\partial}{\partial b}\int_0^\infty (H(b-q)\cdot(v-q)+G_0)p(q)dq \\ &= \frac{\partial}{\partial b}\left[\int_0^b (v-q)p(q)dq + G_0\int_0^\infty p(q)dq\right] \\ &= \frac{\partial}{\partial b}\int_0^b (v-q)p(q)dq \\ &= (v-b)p(q=b)\end{aligned}$$

したがって、 $\mathbb{E}_q[G(b,q)]$ が最大となるための条件は $b = v$ ある。

REFERENCES

A. K. Agogino and K. Tumer. QUICR-learning for multi-agent coordination. AAAI'06, 2006.

A. Dosovitskiy and V. Koltun. Learning to act by predicting the future. *ICLR'17*, 2016.

G. M Edelman. *Neural Darwinism: The theory of neuronal group selection.* Basic books, 1987.

J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients. *arXiv:1705.08926*, 2017.

David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

Andrew V Goldberg, Jason D Hartline, Anna R Karlin, Michael Saks, and Andrew Wright. Competitive auctions. *Games and Economic Behavior*, 55(2):242–269, 2006.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Venkatesan Guruswami, Jason D Hartline, Anna R Karlin, David Kempe, Claire Kenyon, and Frank McSherry. On profit-maximizing envy-free pricing. In *ACM-SIAM symposium on Discrete algorithms*, 2005.

Eric A Hansen, Daniel S Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, volume 4, pp. 709–715, 2004.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *ICLR'16*, 2015.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Roger B Myerson. Mechanism design by an informed principal. *Econometrica: Journal of the Econometric Society*, pp. 1767–1797, 1983.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

S. Sukhbaatar, R. Fergus, et al. Learning multiagent communication with backpropagation. In *NIPS'16*, 2016.

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pp. 1058–1066, 2013.

Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv:1511.06581*, 2015.

T. Zahavy, N. Ben-Zrihem, and S. Mannor. Graying the black box: Understanding DQNs. In *ICML'16*, 2016.