NEURON AS AN AGENT

Anonymous

ABSTRACT

We propose *Neuron as an Agent* (NaaA) as a novel framework for reinforcement learning (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. 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. 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. 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.

1 INTRODUCTION

Deep reifnorcement 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 from Atari, 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 $N_i^{\mathrm{out}} = \{j | (v_i, v_j) \in \mathcal{E}\}$, and a set of units which unit i is connected from is $N_i^{\mathrm{in}} = \{j | (v_j, v_i) \in \mathcal{E}\}$. We denote $N_i = N_i^{\mathrm{in}} \cup N_i^{\mathrm{out}}$.

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.

4.1 Cumulative Discounted Profit Maximization Framework

We denote the external reward which unit v_i receives at time step t as R_{it}^{ex}, where $\sum_{i=1}^{n} R_{it}^{\mathrm{ex}} = R_t^{\mathrm{ex}}$ holds. From N3, reward R_{it} which v_i receives at t can be written as following:

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

The equation devicded into positive terms and a negative term, we name former as revenue, and latter as cost, and denote them $r_{it} = R_{it}^{\mathrm{ex}} + \sum_{j \in N_i^{\mathrm{out}}} \rho_{jit}$, $c_{it} = \sum_{j \in N_i^{\mathrm{in}}} \rho_{ijt}$, respectively. We name R_{it} as profit.

In this case, v_i maximizes the cumulative discounted profit G_{it} represented as the following equation.

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

G_{it} is unobserved unless the time is reached at the end of the episodes. Since prediction based on the current value is needed to select the optimal actions, we approximate G_{it} with value function $V_i^{\pi_i}(s_{it}) = \mathbb{E}_{\pi_i}\left[G_{it} \mid s_{it}\right]$. In this case, the following equation holds.

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

Hence, we only have to consider maximization of revenue and value function and minimization of cost. $R_{it} > 0$, namely $r_{it} > c_{it}$ indicates that the unit give the additional value to the obtained data. If $R_{it} \leq 0$ for all t, the unit acts NOOP since $V_i^{\pi_i}(s_{it}) \leq 0 < \delta$.

TODO: V の式の正しさを検証する。

5 Optimization

To maximize cumulative discounted profit in a framework of NaaA, it is important to balance the two contradicting criteria, revenue r_{it} and cost c_{it} To achieve that, we employ mechanism design.

TODO: 否定的な結論。何が問題か。

The reason why we introduce mechanism design is, unlike the several existing studies (Sukhbaatar et al., 2016), NaaA assumes all the agent is not cooperative but selfish. (TODO: このあたりの文章は要検討) If we naively optimize the optimization problem of NaaA, we obtain the trivial solution that the internal rewards will converges to 0, and all the units becomes NOOP. Hence, the multi-agent system should select the action without any information, and it is equivalent to take an action randomly. Therefore, the external reward R_t^{ex} shrinks obviously.

5.1 Envy-free Auction

To achieve the Pareto optimality, we borrow the idea from digital goods auction. The auction theory belongs mechanism design, and it towards to unveil the true price of the goods. Digital goods auction is one of the mechanism from auction theory, and it is target to copyable goods without cost such as digital book and music.

Although there are several variation of digital goods auction, we use envy-free auction (Guruswami et al., 2005) because it requires simple assumption. The assumption is same goods have one price at the same time. In NaaA, it can be represented as the following assumption:

N5: (Law of one price) If $\rho_{j_1,i,t}, \rho_{j_2,i,t} > 0$, then $\rho_{j_1,i,t} = \rho_{j_2,i,t}$.

It means that v_i has an intrinsic price at the same timing t. We denote the price as q_{it}.

We show the process of envy-free auction in left of Figure 1. It shows the negotiation process between one unit in sending activation and a group of units which buy the activation. The negotiation performed per time step in RL. We name the unit in sending activation as a seller, and units in buying activation as a buyer. First, the buyer bid the unit in bidding price b_{jit} (**1**). Next, the seller decides the optimal price $\hat{q}_{it}$, and perform allocation (**2**). After allocation, the buyers perform payment as $\rho_{jit} = g_{jit}\hat{q}_{it}$ (**3**), and the seller only send the activation x_i to the allocated buyers (**4**). The buyer which cannot receive the activation approximates x_i with $\mathbb{E}_{\pi}\left[x_i\right]$.

In the following, we discuss of revenue, cost and value function based on Eq:(3).

Revenue: The revenue of a unit is given as the following equaiton:

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

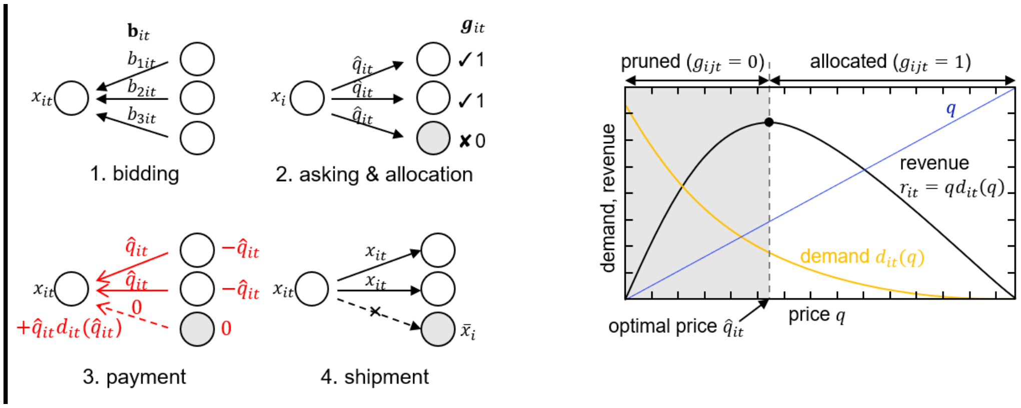

Figure 1: **Left**: the process of trade in envy-free auction. **Right**: a price determination curve for a unit. Revenue of a unit is a product of monotonically decreasing demand and price, and the price maximizing the revenue is the optimal price.

where $g(\cdot, \cdot)$ is allocation, and defined using a step function $H(\cdot)$ as $g(b, q) = H(b - q)$. $d_{it}(q_{it})$ is a count of units which the bidding price for q_{it} is more than or equal to q_{it}, and named as demand. q_{it} maximizing the equation is named as the optimal price, and denoted as $\hat{q}_{it}$. As the second term in the equation is independent of q_t, the optimal price $\hat{q}_{it}$ is given as the following euqaiton.

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

We illustrate the curve of q_{it} in the right side of Figure 1.

Cost: The cost is internal reward which the unit should pay to other units, and it is represented by the following equation:

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

Although c_{it} itself is minimized when $b_{ijt} = 0$, this has trade-off with the next value function.

Value Function: Value of the value function $V(s_{i,t+1})$ depends on $s_{i,t+1}$. As we already defined, the internal environment of v_i is a set of connected units, and the output of units affect to evaluation from the units, namely, weight of edges. As the learning rule of a typical artificial neural network obeys to law of Hebb, the reward becomes lower because weight of unit which do not contribute the accuracy of output becomes lower. Hence, if we minimize b_{ijt} and let $b_{ijt} = 0$, then the purchase of activation fails, the reward can the unit can obtain from the units which the unit connect to becomes lower in the future.

Then, we denote the allocation as $\mathbf{g}_{it} = (g_{i1t}, \ldots, g_{iNt})^{\mathrm{T}}$, and consider effect for value function in the cases when an unit succeed to purchase v_j or not. The value function can be written as the equation using state-value function $Q(s_{i,t+1}, \mathbf{g}_{i,t+1})$.

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

We name $o_{ijt} = Q_i^{\pi_i}(s_{it}, \mathbf{e}_j) - Q_i^{\pi_i}(s_{it}, \mathbf{0})$ as *counterfactual return*, which is equivalent to cumulative discount value of counterfactual reward (Agogino & Tumer, 2006). That is, the cost the unit will pay is $\hat{q}_{it}$ in success of purchasing data, and o_{it} otherwise.

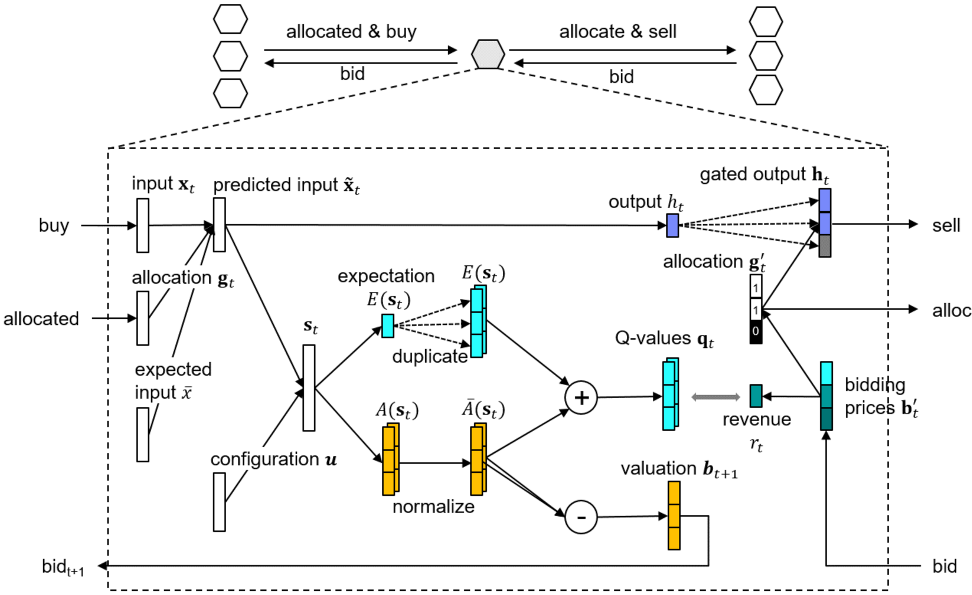

Figure 2: Valuationn Net は情報の価値を評価し、bidding price を決定する。下位のニューロンに対して入札し、信号を購入する。購入したデータを用いて、データを次のニューロンに対して売る。

Therefore, the optimization problem is below.

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

Note that we take expectation $\mathbb{E}_{\hat{\mathbf{q}}_t}[\cdot]$ because the asked price $\hat{\mathbf{q}}_t$ is unknown for v_i except of $\hat{q}_{it}$, and $g_{iit} = 0$.

Then, how much is bidding price b_{it} to maximize return? The following theorem holds.

Theorem 5.1. *(Truthfulness) the optimal biding price maximizing return is* $\hat{\mathbf{b}}_{it} = \mathbf{o}_{it}$.

See Appendix for the proof.

That is, the unit should only consider its counterfactual return (!). Hence, in the mechanism of NaaA, the unit obeys as if performing valuation to the other units, and declare the value truthfully.

Then, the corollary holds,

Corollary 5.1. *The Nash equilibrium of an envy-free auction* $(\mathbf{b}_{it}, q_{it})$ *is* $(\mathbf{o}_{it}, \operatorname{argmax}_q q d_{it}(q))$.

5.2 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。

6 Experiment

6.1 Single-agent environment

6.2 Multi-agent environment

6.2.1 ViZDoom

To 阿久澤君: ここの執筆をお願いできないでしょうか

7 Discussion

7.1 Disadvantage

The one of disadvantage is computational complexity. As envy-free auction uses sort operation for computing demand, several part should be serialized. It should be improved with Approximation.

For optimization method, although envy-free auction guarantees truthfulness if the prices of buyer are sealed, in the case which buyer can communicate each other and shares price information, the buyer can fake the price with lower demand in collusion. For the issue, several solution such as random sample auction Goldberg et al. (2006) is proposesd.

Adoptive dropconnect has difficulty for implementation for several neural network. Although, we published source code on GitHub for Linear and CNN, implementation for RNN is a future work.

7.2 Application

NaaA can be applied on learning distributed environment on computer network such as peer-to-peer network, and controlling sub-module of robot such as multiple camera. Specifically, it can be applied to various method as below.

- Hyperparameter tuning. Several algorihtm are already proposed such as neuroevolution using genetic algorithm. In the case, profit or counterfactual return can be used to fitness function.
- Pruning. Reducing computing cost with downsizing neural network.
- Attention control. In a part of research of attention, they are using reinforcement learning to control attention.
- Ensamble. Our method can be applied to mix multiple models.

These application is direction of the research.

8 Conclusion and Future Works

This paper proposed NaaA, a reinforcement learning framework which treats each units on a neural network as an agent. First, we pointed out there are dilemma problem if we naively optimize NaaA, and proposed the optimization method with auction. As the result, the action which the units evaluates counterfactual return of other units is obtained as Nash equilibrium. Besides, we proposed Q-learning based algorithm, adaptive dropconnect, to dynamically optimizing the topology of neural network with evaluating counterfactual return. In the evaluation, we performed experiments based on single- and multi-agent platforms, and showed our experimental result improves existing methods.

As future direction, we use on-policy method to perform adaptive dropconnect, and considering application combining genetic algorithm.

Appendix

A.1 Proof of 5.1

As for a buyer, asking price q for a seller is unknown,[we address q which have support $[0, \infty)$, and consideri to maximize $\mathbb{E}_q\left[G(b, q)\right]$, In this case, the following equation holds.

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

, Therefore, the condition to maximize $\mathbb{E}_q\left[G(b, q)\right]$ is $b = v$.

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.