Multi-Agent Communication

MVA — Reinforcement Learning: Final Project Report

Daniil Lotkov
ENS Paris-Saclay

Abstract

Multi-agent approach has shown to be an effective method of solving both the tasks that require cooperation, and that are classically considered to be individual. Speech and communication is often a necessary prerequisite to the effective cooperation. This work presents an evaluation of Mordatch & Abbeel (2017) [6] paper that introduces agents to 2D simulated physical environment that has a goal of reaching the given landmarks while utilizing communication to share the information about them. Contributions of this work include modifying the open-source implementation with visualization, automated experiments, and additional entropy-based losses. The default vocabulary frequency-based loss is shown to be marginally effective and the newly introduced entropy loss, while being unstable for not carefully selected normalization dimensions, consistently outperforms standard word penalization with a right choice of the entropy calculation method.

1. Introduction

By capturing statistical patterns in large corpora, machine learning has enabled significant advances in natural language processing, including in machine translation, question answering, and sentiment analysis. However, for agents to intelligently interact with humans, capturing the statistical patterns is insufficient. The reason for that is that approaches that learn to plausibly imitate language from examples, while tremendously useful, do not learn why language exists. Such supervised approaches can capture structural and statistical relationships in language, but they do not capture its functional aspects, or that language happens for purposes of successful coordination. Evaluating success of such imitation-based approaches on the basis of linguistic plausibility also presents challenges of ambiguity and requirement of human involvement. In this paper we study the work of Mordatch & Abbeel (2017) Emergence of Grounded Compositional Language in Multi-Agent Populations [6]. We formulate the plausibility of the multi-agent approach, develop an understanding of why the communication is beneficial, and evaluate the algorithm using different metrics and parameters.

2. Problem analysis

Multiple questions arise when analyzing the subject of multi-agent communication setups used in solving tasks. First, do we need communication? Second, should the agents be implemented as a diverse set of autonomous actors, or should they be coordinated by a central controller? Third, if the agents are autonomous, what kind of communication is necessary for them to cooperate effectively in the task? We will follow through these questions in the following subsections.

2.1. Multi-agent approach justification

Let us unravel the rationale behind the use of multi-agent setups from the least strong assumptions to the most strong ones.

2.1.1 Standard environments

In certain tasks such as pursuit and evasion, multiple agents must coordinate their behavior to achieve a common goal. The study on the viability of a multi-agent approach that utilizes neural networks can be traced back to as early as 2001 with Cooperative Coevolution of Multi-Agent Systems by Yong and Miikkulainen (2001) [7]. It studies multiple agents that were trained together versus the ones that were trained separately and then united in the test time on the prey-predator task.

Since the speed of the prey in the given environment is higher than that of the other entities, as expected, the capture task could be successfully achieved only if the hunters were trained in a cooperative manner. The Fig. 1 demonstrates that an individually trained agent is not capable of completing the task, but the agents that were trained in a multi-agent manner, successfully achieve the goal.

While this result can seem trivial, the following question that build up on that are more complex. Authors also study

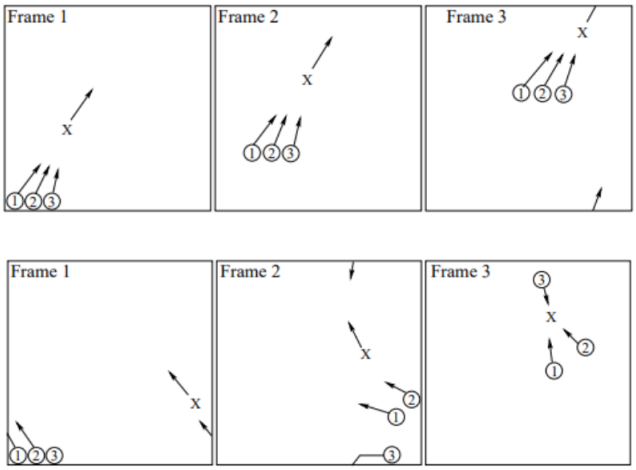

Figure 1. The enclosed tor-shaped prey-capture environment. Top series correspond to the individually-trained agents, bottom series display the multi-agent training setup result.

the effect of training scheme choice, i.e. centralised vs. decentralised controller.

You can see an illustration of both types of controllers on Fig. 2. The paper concludes that the single neural network that controls all the agents is inferior to the individual agent models. Also, a communicating group was shown to be more effective in the above-mentioned paper, by pointing out that it is more robust to changes in the environment. On average, the coevolution of the three neural network controllers was almost three times as fast as the evolution of a centralized controller in finding a reasonable solution. Furthermore, the single neural network was unable to evolve to the level of expertise required to solve all nine benchmark cases within the 400 evolutionary cycles, whereas the cooperating neural networks were able to do it every time.

2.1.2 Non-conventional tasks

While this result is intuitively expected in terms of an environment posing a task that has a natural need in coordination, the paper of Havrylov & Titov (2017) [3]. They study the results of an agents communication on the setting where two agents engage in playing a referential game (image classification). Authors observe that agents, from scratch, develop a communication protocol necessary to succeed in this game. The protocol agents induce exhibits a degree of compositionality and variability (i.e. the same information can be phrased in different ways), both properties characteristic of natural languages.

The Fig. 3 shows some samples from the MSCOCO 2014 validation set that correspond to (5747 * * * *) code. Images in this subset depict animals. On the other hand, it seems that images for (* * * 5747 *) code do not correspond to any predefined category. This suggests that word order is

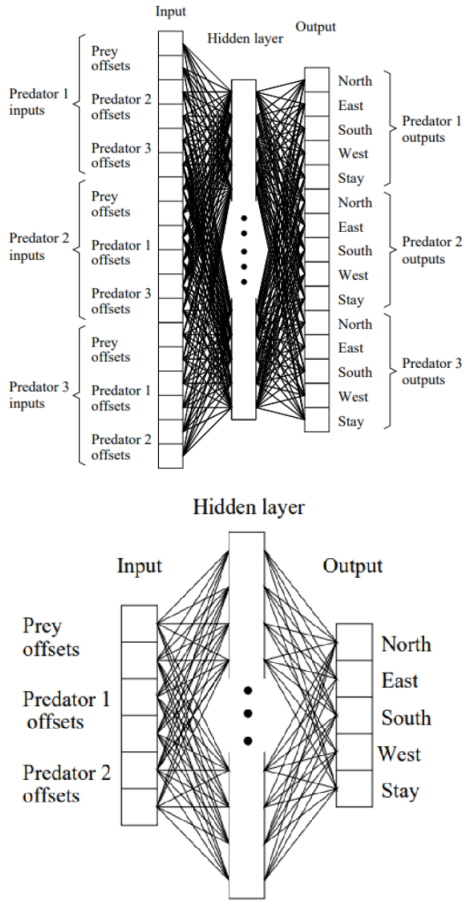

Figure 2. Two types of controllers used (centralised and decentralised).

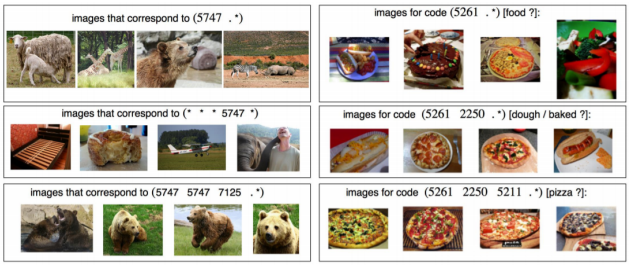

Figure 3. The samples from MS COCO that correspond to the particular code.

crucial in the developed language. Particularly, word 5747 on the first position encodes presence of an animal in the image. The same figure shows that message (5747 5747 7125 * *) corresponds to a particular type of bears. This suggests that the developed language implements some kind of hierarchical coding. This is interesting by itself because the model was not constrained explicitly to use any hierarchical encoding scheme. Presumably, this can help the model efficiently describe unseen images. Nevertheless, natural

language uses other principles to ensure compositionality. The model shows similar behaviour for images in the food domain.

3. Related work

Many papers have studied communication in the multi-agent setting, but the particular case this research is focused on is a physically simulated environment with particular objectives. It is argued in [6] that the physical environment facilitates more complex interactions and, thus, more complex "speech". Some papers have been considered to be a subject of the study, including ones that operate in physical environment: Lowe *et al*. (2020) [5], Blumenkamp & Prorok (2020) [1], and the ones that don't: Foerster *et al*. (2016) [2] and Lazaridou *et al*. (2017) [4]. However, the choice was made to prioritize several parameters: first, environment should be physically simulated, second, at leas some part of the code should be publicly available, third, the result should be easy to view and interpret. The best paper that correspond to these criteria was the aforementioned Mordatch *et al*. (2017) [6], which was selected as the object of the study.

4. Methodology

The following subsections present the architecture of the studied paper and the contributions made, along with the associated losses and training methods.

4.1. Environment

Following the rationale of having a complex enough environment for agents to develop sophisticated communications, the physically simulated medium is selected. This environment is a constrained 2D space that contains N movable entities and M static landmarks with no collision detection enabled. All the entities in this space posses a physical characteristics such as shape and color (that are sampled from the limited set every epoch), while agents can perform actions of moving continuously and emitting a "speech" vector. Position and the physical characteristics of each agent is given by a vector x, and speech vector has a size of K and codes the emitted symbol c_i intensity in the value of the respective dimension. Each agent has its internal goals specified by a vector g that is computed upon other agents utterances and their physical characteristics. To aid the agents and to ease the models of keeping the memory of the previous time steps, each agent is equipped with an internal memory queue of size m that is updated every step. The full state of the environment is given by $s = [X_{1,\dots,(N+M)}, C_{1,\dots,N}, m_{1,\dots,N}, g_{1,\dots,N}]$ with an individual agent observation $o_i = [X^i_{1,\dots,(N+M)}, C^i_{1,\dots,N}, m_i, g_i]$ where X^i_j represents the physical state of an agent j perceived from the ith agent point of view. The environment visualization for the non-trained agents is presented at the Fig. 4.

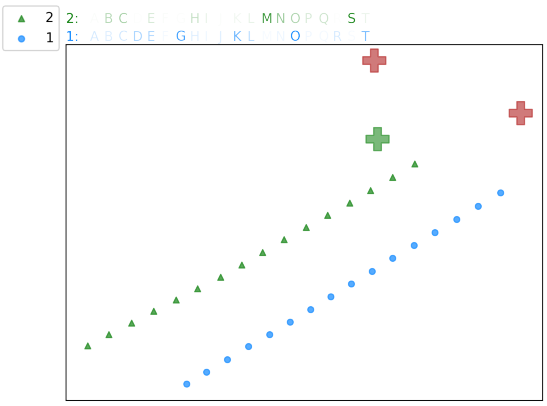

Figure 4. Agent trajectories, landmarks and utterances at the final time step. Alpha channel of the symbols represent the sum of the agent's utterances over the epoch.

4.2. Policy

Agents act in accordance with the stochastic policy π that is shared across all agents and is parameterized by the set of parameters θ. The authors choose to use the end-to-end differentiable neural network model. They utilize the Gumbel-Softmax approach for keeping the communication medium differentiable. Fig. 5 describes the policy architecture that an agent utilizes. Utterances c and physical observations x are first processed by GRU units and then passed to the fully connected layer. Then, the features that correspond to different symbols and entities are filtered by a maxpooling operation and concatenated with the goal prediction g to produce agent utterance and movement action. Each step of feature extraction is also aided with the memory bank information to facilitate the model learning process.

4.3. Rewards

The objective function of an agent is designed to work with the given finite horizon task, and comprised of several different components that allow it to both achieve goal understanding, move itself in the space, and put some constraints on the utterances it uses.

4.3.1 Physical

The first and the most fundamental to the task loss function component is a distance between the goal landmark and the selected agent. The authors use the square of Euclidean distance for defining the cost of the final agent position that can be expressed by:

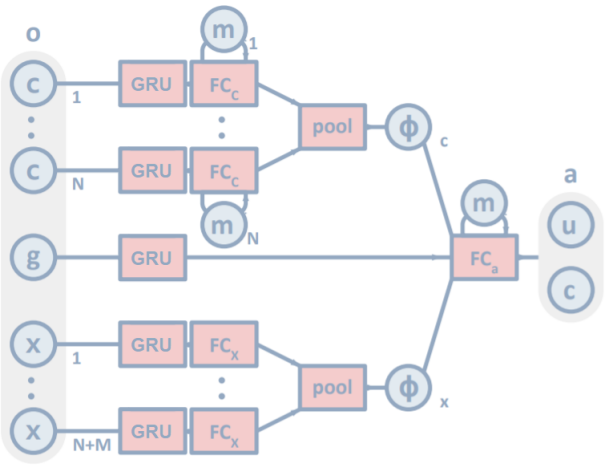

Figure 5. Policy architecture (modified from [Mordatch & Abbeel, 2017]). Features are extracted using a GRU cell and a fully connected layer and then prioritized by the pooling layer.

$$r_d = \sum_{t=1}^{T} \sum_{i=1}^{N} ||d_{i,t} - d_{g_i,t}||^2$$

where $d_{i,t}$ is a position of ith agent at timestep t and $d_{g_i,t}$ is a relative position of a goal assigned to this agent.

4.3.2 Goal prediction

In order to facilitate the convergence the algorithm, another loss function is used. During training, agents predict goals of other agents, however they do not use it as an input to the model. The cumulative value for this loss has the form of:

$$r_g = \sum_{\{i,j|i \neq j\}} ||\hat{g}_{i,j}^T - g_j^T||^2$$

where $\hat{g}_{i,j}^T$ is an estimate of jth agent goal given by ith agent at the final timestep, and g_j^T is the ground truth value.

4.3.3 Word frequency

To get the resulting communication protocol to posses some particular qualities, authors use the loss function component that rewards agents using symbols they already frequently use. This component is making already popular symbols more popular, therefore encouraging model to make the dictionary more sparse. Each communication symbol has a probability of being a symbol c_k of

$$p(c_k) = \frac{n_k}{\alpha + n - 1}$$

where n_k is the number of times symbol c_k has been uttered and n is the total number of symbols uttered. α is a hyperparameter that corresponds to the probability of observing a new word. The resulting reward is derived this way:

$$r_c = \sum_{i,t,k} \mathbb{1}[c_i^t = c_k] logp(c_k)$$

However, the implementation that was used did not utilize the logarithm, but since there is no non-linear operations it is functionally similar to the one suggested in the paper.

4.3.4 Utterance entropy

Entropy measure is a useful way of getting an understanding of a distribution characteristics, but there is no single way to compute it over a multidimensional tensor. What defines the final value is whether we treat the given data as a whole distribution, and if not, which dimensions are chosen to be the independent components. Normalization is performed only in the second case and it depends on the choice of dimensions. The definition of entropy of distribution p is $E = -plogp$. In the experiments, both the normalized version and the version without prior normalization or dimension prioritization are used.

In the first case entropy computation is performed the following way:

$$r_e = -\sum_i S' log S', \quad S' = \sum_{j \in N, j \neq i} S_j, \quad S \mid \sum_i S = 1$$

where S is a tensor of N dimensions, and is normalized to 1 on the ith dimension for it to represent the probabilities. All the other dimensions are treated as independent components. To compute entropy of a given tensor without outlining any particular components we do not normalize and do not perform summation before the computation, so that $S' = S$. Further, in the experiments section it will be shown how does the calculation process affect utterance structure, and what particular type of normalization should be used.

5. Experiments

This section present the experiments that were performed on the algorithm and the analysis of the obtained results.

5.1. Word frequency penalization

The first set of experiments was performed on the loss component that was responsible for rewarding the agent for using those words, which were already dominant in its vocabulary. We also study the effect of the α constant. On the Fig. 6 you can see the effects of enabling non-frequent word

penalization on the vocabulary used. The figure demonstrates that the loss indeed makes the distribution of the symbols used more spiked on the subsection of symbols rather than being relatively uniform.

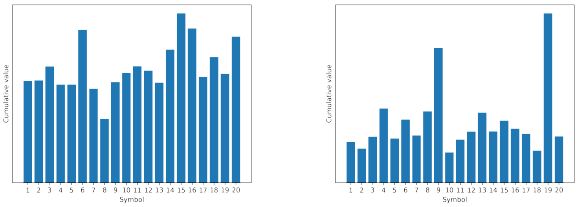

Figure 6. The default distribution of utterance vectors (left) and the distribution with word frequency penalization loss enabled.

We can also observe the effect of the α on the final distribution on the Fig. 7.

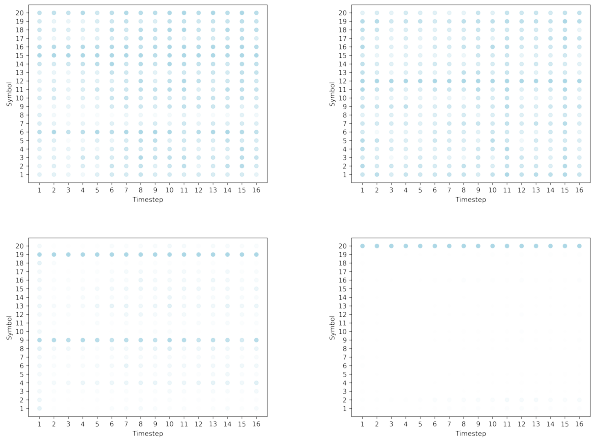

Figure 7. The distribution of utterance symbols over timesteps depending on the word penalization $alpha$ (None, 0.2, 0.5, 2.0).

We will illustrate the difference between frequency penalization and entropy penalization in the following subsections.

5.2. Vocabulary size

The vocabulary size parameter is set to be 20 by default, but for the environment that comprises of no more than 5 entities and simple goals, such a high dimensionality seems to be excessive. The Fig. 8 shows the distribution of an emitted symbols.

We can observe that the symbol distribution always possesses some non-uniformity, and always saves approximate amount of peaks, and the expected entropy of all this system should be the same in case of no information transferred though the channel. However, on the Fig. 9 it can be seen that despite the fact that standard deviation decreases with the dimensionality of the system, as it should be, the entropy

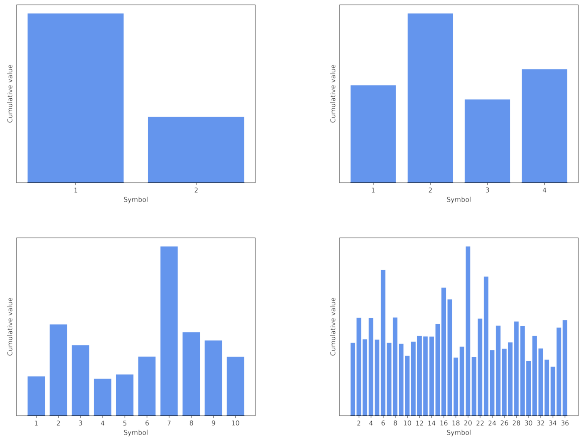

Figure 8. The distribution of utterance vectors sum on the vocabulary size (2, 4, 10, 36)..

rises, which indicates the fact that the amount of information transferred stays approximately the same and is diluted in the extra dimensions.

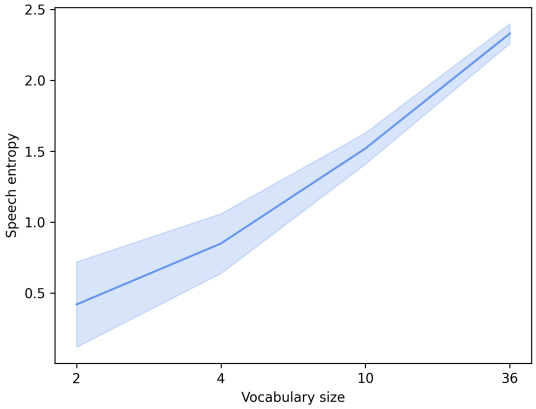

Figure 9. Utterances entropy value on vocabulary size dependency.

5.3. GRU vs LSTM

Different recurrent processing modules has also been studied. Original paper does not contain the recurrent modules and relies on a linear feed-forward layers, however the implementation I used makes use of GRU, and also LSTM units were added to make a comparison. The Fig. 10 demonstrates example of a run both of these techniques.

We can see that the performance of these two examples are marginally similar expect for LSTM showing a little bit more stability in the final point of the trajectory, while GRU utilizing agent tends to oscillate after reaching the landmark.

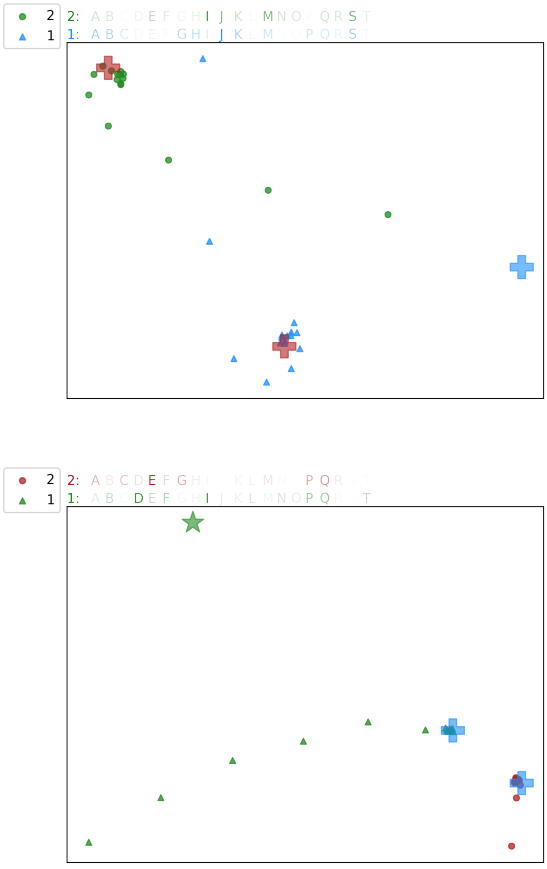

Figure 10. Agent trajectories, landmarks, and emitted symbols of GRU (left) and LSTM (right).

5.4. Entropy penalization

As stated in the sections above, entropy can be computed by different ways, that is defined by which dimensions are being adopted as independent, and which dimension is a distribution dimension itself. Now we will compare symbol structure while using agent, batch or symbol dimension as primary ones to computing entropy without prior sum or normalization.

From the Fig. 11 we can see that the only rewards that have not produced a degenerate distribution are non-normalized and batch-normalized versions. An explanation to the failure of agent and symbol dimensions normalization is in the fact that if we minimize the entropy between agent's utterances, it is likely to make one of the agent stop producing utterances at all, and does not force constraints on an individual agent. On the contrary, symbol dimension entropy, as expected, produces singular distribution due to the reward given for the lowest entropy of symbols emitted. However, while the case of computing entropy without selecting primary dimension and normalization have no

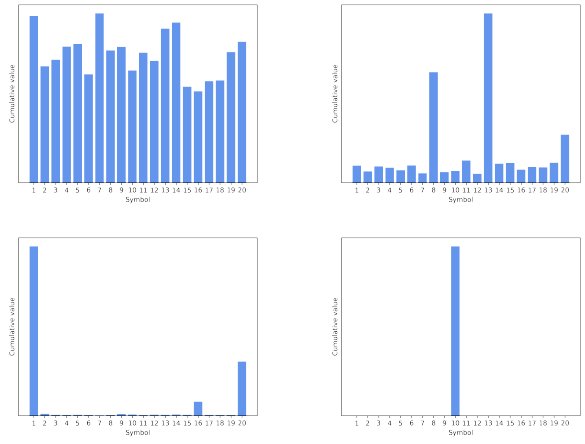

Figure 11. The distribution of utterance vectors sum on the entropy calculation type (agent dimension, prior-log sum, batch dimension, symbol dimension).

explanation in this paper, the batch dimension entropy produces good results and has an explanation behind. Each batch is comprised of two agents that have random properties and are placed on a random index. When we penalize the entropy using in the batch dimension we force the agents to use different from each other symbols (that also don't repeat for each individual agent), but we do not force entropy to be small over symbols overall, so agents can use several of them with the restriction of not emitting the others.

On the Fig. 12 you can see an entropy evolution of the given methods over epochs.

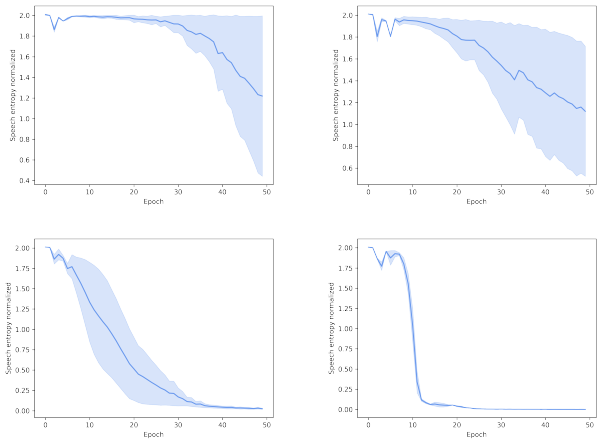

Figure 12. Entropy of utterance vectors (agent dimension, prior-log sum, batch dimension, symbol dimension).

We can observe the instability of the methods that does not rely on sensible choice of a primary dimension. Agent dimension entropy calculation is unstable as well as the verion that does not ustilize any prior-log summation. Symbol dimension entropy loss is the most predictable, but it yields

a singular utterance distribution. However, batch dimension entropy calculation represents the best option, which shows a good performance combining it with some sensible margin of stability.

6. Conclusion

In the scope of the following work the existing open-source algorithm implementation was debugged, provided with a data logging, visualization and experiment automation instruments, and augmented with additional rewards. The default vocabulary size was shown to be excessive for such a task with smaller vocabularies achieving the same results, while containing more information. Also, an LSTM usage has been tested which appeared to give no significant difference except for less oscillation at the point of reaching the landmark. The default existing technique of vocabulary constraints showed itself to be marginally effective managing to highlight frequent symbols, but failing to reduce the utterance noise. However while depending on a primary dimension choice, the new entropy-based reward function demonstrated good performance in terms of vocabulary optimization. Further research on this article can include the influence of the entropy loss on different vocabulary sizes and study of the optimal symbol space dimensionality.

References

[1] Jan Blumenkamp and Amanda Prorok. The emergence of adversarial communication in multi-agent reinforcement learning. 2020.

[2] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *CoRR*, abs/1605.06676, 2016.

[3] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *CoRR*, abs/1705.11192, 2017.

[4] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *CoRR*, abs/1612.07182, 2016.

[5] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *CoRR*, abs/1706.02275, 2017.

[6] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *CoRR*, abs/1703.04908, 2017.

[7] Chern Han Yong and Risto Miikkulainen. Coevolution of role-based cooperation in multi-agent systems. *IEEE Transactions on Autonomous Mental Development*, 1:170–186, 2010.