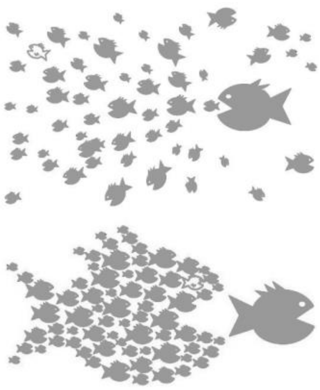
Spring 2020

Internal Badger Workshop

Session 5: Psychology of Badger (**Intro- vs. Extro-version**)

Moderator: Jan Feyereisl



This session is all about the **benefits**/drawbacks of **collective vs. solitary learning & computation**. We see and argue a great many benefits in learning and computing collectively, but this is not necessarily backed up yet by results or explicit theories and formulations that could guide us on how to achieve those in practice in Badger. How can we correct this?

Pre-discussion Comments & Resources

The main purpose of this session is to have a discourse on **what is the true benefit of the collective nature (multi-agentness) of learning and computation** that we argue for in Badger. In other words, should a Badger agent be a loner, an introvert, i.e. a single expert, just like its biological counterpart or rather a more social and extroverted collective, maybe much like the bees whose honey Honey Badgers so dearly love.

As all of this occurs inside the mind of our agent, we entitled this session "Psychology of Badger". Even more precisely, we might have wanted to call it "Social Psychology of Badger", but as none of us are Psychologists (as far as I know), correctness of the title is not important. What's more important is maybe whether the field of Psychology and Social Psychology reveals something about the problems we try to tackle (see section below). But maybe that is something that could be resolved once we have some Psychologists amongst ourselves. For me, the most favourable outcome of this session would simply be some ideas and insights on **how and where to show that the collective learning and computation within our agents brings some benefits over learning and computation in a single monolithic agent**.

continued...

Benefits of Collective Computation - Some Suggestions

In our paper, we outline a number of benefits of collective learning and computation, but these are not necessarily exhaustive. Here is an outline of some of the claimed benefits worth talking about:

- **Scalability** - learning a communication policy that defines local interactions among badger experts, should allow for easier and more sensible scalability
    - scalability to more complex inputs (memory)
    - scalability to more complex computation (collective strategy)
    - scalability to more complex problems (memory & strategy)
    - computational scalability - easier distributed computation
    - what other types of scalability are there that are relevant?
- **Extensibility** - modularity might enable easier extensibility that could avoid existing issues present when extending standard deep learning systems
    - Avoiding catastrophic forgetting
    - On demand computation - adding experts on the fly
    - Growing/Graduality of an agent
- **Others?**

These are just some potential benefits, but **there are definitely more benefits of collective computation out there. What are they? And how do they apply and help in our case?**

Having outlined, and possibly discussed the above during the session, how can we turn the above benefits and formulate them into specific:

1. **learning procedures**
2. **loss functions**
3. **other things relevant for Badger and more broadly?**

that enable our agent to learn a policy that is much more likely to learn such collective behaviour within which the above would emerge?

Examples:

1. In outer loop learn with varied numbers of experts (Microbadger-like)
2. Loss calculated from relative improvement of adding an expert
    a. Inner loop with 1 expert, then 2, then 3, then *n*
    b. Maximize the relative benefit of having more experts
3. Maximizing benefit of local expert interaction (+ inner loss)
4. others...?

Back to Social Psychology

Looking back at Social Psychology, one might find some loose connections and interesting snippets that might be worth pondering about. For example, in "Group Dynamics" a group, defined as two or more individuals that are connected by social relationships, is distinguished from aggregates, a term used for things consisting of elements but considered as a whole. A group, in this sense, has a number of emergent properties:

- **Norms** - *a form of API that's understandable by individuals (implicit rules and expectations)*
- **Roles** - established specializations that are beneficial to the collective
- **Relations** - patterns of liking within the group (success, status, etc.)

Interestingly, it is argued that temporary groups and aggregates share few of those, e.g. a bus line. What can we learn from this and is there a computational and a learning insight that could be exploited practically during training of our agent?

But this is just an amateur's snippet of arbitrary analogy. There surely are many more interesting concepts worth exploring. I dare to suggest one more, potentially a very important one for decision making. In the 1970's there was a study on the benefits of collective decision-making by Hall and Watson. They have essentially showed that a group deciding on how to survive on its own is significantly poorer at making decisions than a group that was tasked to use group consensus and in particular **avoid voting, averaging** and other conflict-reducing techniques. The instructed group had to employ strategies and **explanation mechanisms** that in effect had to convince others of why they chose what they chose, resulting in extracting more informative decisions than averaging which is no better than the decision making of a single average participant. I was part of this experiment recently and even though this might be common sense to some, I was astounded how effective it was.

The question is, can we transform the above into a computational or learning procedure to exploit the collective benefits and have our agent reap the collective rewards?

More arbitrary questions and notes to ponder about during the discussion

- **Collective Learning** vs. Collective Computation! - under-appreciated?
    - Teacher-Student
    - Learning Using Privileged Information
    - Schools?
- Example of a team solving an unknown problem - different expertise?
- Train with different number of experts
    - Optimize for better performance with added expert vs single group?
    - Optimizing relative improvement of adding experts?
- Relative comparison of a Badger agent/expert and a neuron/RNN?
    - How do they differ?
    - What implications does this have on connectivity, learning (optim), etc.?
- The concept of "Parasites"
- Micro-badger concept
- Other areas that deal with collective computation, multi-agent setting, etc.

- Multi-agent reinforcement learning
- Learned Routing Networks & Modular Computation Approaches
- Bio-inspired methods - Ant-Colony, Swarm, Herding, others?

References

Adjodah et al., (2020), *Leveraging Communication Topologies Between Learning Agents in Deep Reinforcement Learning*, https://arxiv.org/abs/1902.06740

Clemens et al. (2019), *Routing Networks and the Challenges of Modular and Compositional Computation*, https://arxiv.org/abs/1904.12774

Hall, J., & Watson, W. H. (1970). *The effects of a normative intervention on group decision-making performance. Human Relations*, 23(4), 299–317. https://doi.org/10.1177/001872677002300404

Krakauer, D., Bertschinger, N., Olbrich, E. et al. *The information theory of individuality*. Theory Biosci. (2020). https://doi.org/10.1007/s12064-020-00313-7

Discussion Notes

- There might be a way to represent Badger within an LSTM via block diagonal matrices
    - this might be beneficial for comparison of badger and LSTM at a comparable level of computation capacity (keeping it fixed)
    - also this could help offer a viewpoint on Badger from the lens of existing and known machine learning methods and architectures
    - when comparing, how do we avoid artificially disadvantaging the monolithic baseline?
    - there might be a benefit in taking an adversarial point of view
- Would doing the opposite of example 2 loss?
    - Does this have relationship to dropout?
- Set up the loss/learning procedure as a GAN-like game where the biggest improvement is sought, looking at one expert vs. two experts
- Bipartite badger might also be able to add something to the mix with nodes calculating losses and synapses the relative benefit of two connecting nodes
- Parasite - a unit that somewhat externally observes statistics of a badger agent and aims to improve those statistics by interfering with the host in some way
- Can novelty search be somehow helpful for this topic?
    - Ensembles in badger are not beneficial/necessary
    - Niches are maybe much more desirable
- NASA Psych experiment (Hall and Watson 1970)
    - It is less about diversity of participants/experts and more about deliberation that is convincing and weeds out the thoughts that are less reasoned/research/argued.
    - Research likely already done on something similar in swarms

            - weighing strategy might help
            - it is hard to avoid averaging
            - make averaging hard/costly
    - It is difficult to constrain what is and what is not learned
        - how can we avoid this issue?
    - Use of prediction markets in this scenario
        - in this case the signal about which experts reason is too limited/sparse and simple, whereas in the NASA experiment the discourse process is significantly more complex.
- Multi-agent systems can self-organize (contrary to monolithic systems)
- Cognitive processes may require multi-agent level of separation, parallel processes, temporary spawned worker threads, arbitrary time-scales, replication and redundancy, agents competing for limited resources. It would be harder to get all these properties in a monolithic system.
- It is easier for us to see modules and properties in a multi-agent system
- More natural way of representing skills

References/Researchers mentioned during discussion

- Irving & Askell, "AI Safety Needs Social Scientists", Distill, 2019.
- Agarwal, Schuurmans and Norouzi, An Optimistic Perspective on Offline Reinforcement Learning, NeurIPS 2019 Deep RL Workshop, 2019
- Researchers
    - Iain Couzin
    - Jessica Flack
    - David Krakauer