Spring 2020

Internal Badger Workshop

Session 7: Principia Badgerica

Moderator: Jan Feyereisl

- What are the Badger **principles (foundations)** that we have discovered already?
- What have we **learned** during Badger development so far?
- This discussion should ideally result in a document outlining all the important findings we have made thus far

Pre-discussion Comments & Resources

In this session we should try to lay out all the principles and foundations of badger as well as important, surprising and revealing findings that we have discovered so far.

We can start with outlining the **foundational concepts and principles** of badger.

We can then try to expand and gradually add things to this list and as sub-points.

1. **Modular, Collective, Multi-Agent** - collective learning & computation
2. **Meta-reasoning** - deciding how to think / allocate computation resources
3. **Meta-learning** - learning how to efficiently use data
4. **Learning to Learn** - Meta-reasoning + Meta-Learning
5. **Communication** - homogeneous policy
6. **Hidden state / activations** - substrate of a unit within the collective
7. **Scalability** - generalize and expand in new directions
8. **Locality** - composition of local rules enables scalability
9. **Growing** - infinite continual learning and adaptation
10. **Generality** - learning and adaptation to many envs/problems/tasks
11. **Minimum Viable Environments** - MVEs and their generators
12. **Input and Feedback** - error/feedback on input

Now, we can go over each of the above and discuss important **principles associated** with each, as well as **findings and discoveries** that we have made in each.

Findings and Discoveries

- Difficulty in training badger
- Initialization
- Module collapse - diversity and symmetry breaking
- Challenging nature of guessing game
- What is learned in the inner loop
    - Triangulation
    - Fast convergence when randomizing num. steps
- How hard is it to learn SGD
- Learning of invariances
- Generalization across dimensions
- others…

Misc. and Random Notes

- What are the different viewpoints from which we can see Badger?
    - Multi-Agent Reinforcement Learning system
    - A social collective
    - LSTM with tied weights / block diagonal matrix
    - Experts as executors of programs communicated across the topology
    - **others?**
- "*Since the human mind has only a limited amount of computational resources, it has to allocate them in an adaptive manner to solve complex problems efficiently. Having finite time in which to learn means that humans have to be able to make the most of each piece of data available to them, building and then drawing upon a rich model of the world in which learning takes place.*" [1]
- "*Part of what makes people smart is their capacity to make decisions about how they make decisions*" [1]
- Meta-learning in the sense of above should really be happening in the inner loop, so the outer loop should allow for learning how to meta-learn?

References

[1] Griffiths, T. L., Callaway, F., Chang, M. B., Grant, E., Krueger, P. M., & Lieder, F. (2019). Doing more with less: Meta-reasoning and meta-learning in humans and machines. Current Opinion in Behavioral Sciences, 29, 24–30. https://doi.org/10.1016/j.cobeha.2019.01.005

[2] Rosa et al. (2019), BADGER: Learning to (Learn [Learning Algorithms] through Multi-Agent Communication), https://arxiv.org/abs/1912.01513

[3] Badia et al. (2020), Agent57: Outperforming the Atari Human Benchmark, https://arxiv.org/abs/2003.13350

Discussion Notes

Foundational Principles and Benefits

1. **Modular, Collective, Multi-Agent**
2. **Meta-reasoning** - deciding how to think / allocate computation resources
   - **a.** Deliberation / Variable length computation
   - b. Dreaming
   - **c.** Counterfactuals
   - **d.** *Need for a generative process?*
   - **e.** *Are benefits of collective computation more achievable here?*
3. **Meta-learning** - learning how to efficiently use data
4. **Learning to Learn** - Meta-reasoning + Meta-Learning
   - **a.** inner/outer loop
   - **b.** inner loss / outer loss
   - **c.** learning protocol
5. **Communication** - homogeneous policy
   - **a.** policies
     - **i.** learned
     - **ii.** manual
   - **b.** topologies
     - **i.** hard-wired
     - **ii.** random
     - **iii.** dynamic
   - **c.** messages
     - **i.** algorithm(s)ic snippets
6. **Hidden state / activations** - substrate of a unit within the collective
   - **a.** capacity
   - **b.** representation power
   - c. learnability
7. **Scalability**
   - a. Computational
   - b. Problem
   - c. Life - Infinite inner loop
   - *d. Targeting this in the outer/inner loops*
8. **Locality**
9. **Growing**
10. **Generality**
    - a. Bottlenecks
      - i. Homogeneous communication policy
      - ii. Communication channels?
11. **Minimum Viable Environments (MVEs)**
    - **a.** minimum viable environments and their generators (e.g. Curricula)

12. Input and Feedback

- Badger streamlined our thinking about AGI agents, it has more structure, individual properties should be more testable, we ask better questions than before-Badger, perhaps even enable incremental R&D.
- **Discovery and remembering of new knowledge in the inner loop (vs. "just" an inference in case of e.g. RL^2)**
    a. information bottleneck from outer to inner loop (auto ML, metaGenRL)?
    b. ability to make hypotheses, experiments, extend the knowledge base?

  main research focus:

    - understanding the differences in what inner and outer loop mechanisms do
    - knowledge discovery
    - hypothesis formulation
    - hypothesis testing
- **Distributed and dynamic topology**
    c. still confusion in benefits: efficiency vs. robustness (and extensibility)
    d. example: robustness in distributed databases
    e. sparse connections => faster learning, avoid catastrophic forgetting?

  main research focus:

    - communication (current one not strong enough)
    - negotiation
    - topologies

What have we learned thus far?

- Learning nodes (hidden states) versus learning relationships (weight matrices, communication networks) do different things, have different invariances/directions of guaranteed generalization. Hidden states tend to provide less expressive range than connection networks (see e.g. the mental affordances blog post and followup experiments with the modular version of that). So, during the inner loop, we probably do want connection structures to be learned, not just expert hidden states.
- Going to various kinds of multi-agent architecture doesn't actually have a strong effect on the kinds of solutions that are discovered. LSTMs, Attention Badger, Microbadger, and Bipartite Badger all discover and prefer triangulation as a method for solving the guessing game. So we can't expect the architecture to 'carry us' here by coming up with inherently smarter or more scalable solutions just because of how the network is laid out.

- On the other hand, some kinds of generalization (changing # of problem dimensions in the guessing game) are made possible by architecture choice, and are totally impossible or impractical with other existing network architectures that aren't designed to be extensible.
- So rather than thinking of Badger as something that does something for us, it's more that it's something which lets us do something for ourselves. If we don't make use of that extensibility in the task design, evaluation, or way we deploy it, then the 'multi-agentness' and 'modularity' aspects won't do anything.
- General property of adding more experts in absence of anything else: more experts averages out sources of noise that exist inside the architecture. Therefore the most common scenario to see continuing improvement as # of experts is increased beyond the training range is when you have internal noise, but the improvement is entirely with respect to that artificial difficulty.
- Secondary property of adding more experts is that it seems to make the outer loop training easier or more stable, but this isn't a kind of generalization (since if you then change the # of experts and re-run the inner loop, there's no benefit to doing so in either direction)
- In microbadger, attention badger: stability versus increasing # of experts is fairly easy to achieve by training on some small range of expert count, and this generalizes by factors of 10x or more (stability means that it doesn't get noticeably worse). However stability versus increasing rollout length is always very bad - more than a 20% increase in inner loop rollout length without retraining will generally increase the error, and even with outer loop training often the rollout length has to be increased slowly to prevent training from diverging.
- Attention is somewhat harder to train, but actually in the end things like microbadger with fixed layouts (not even randomly changing the network every step) also get stuck on similar plateaus for the guessing game at least. So the difficulty isn't 'just' attention there.
- Self-supervision during the inner loop can be effective with a small seed of external supervision to get it started, but does seem to require that seed.
- Additional complexity brought in by communication. In the guessing game task, which was found to be solvable using a clever initialization without any communication, adding communication increased the time needed for finding the solution significantly. In part, this could have been expected, as the number of inputs available to experts increase. On the other hand, the expectation was opposite - we though adding communication would make the task easier. Here, instead of helping solve the task, communication seemed to harm it.

Miscellaneous Other Notes

- Difference between choosing an architectural property that is interesting vs. one that is good. Meaning that something that sounds or looks interesting, doesn't necessarily mean that it’s good. One possible example of this is the multi-agent nature of badger. It sounds interesting, but so far it makes learning more difficult than a monolithic agent. But eventually, the multi-agentness might not be good, might even be harmful, but ultimately necessary in order to be able to go beyond what current agents can do.
- One benefit of multi-agentness/modularity is the possible extensibility
- It is important to train the system for extensibility
- There is a difference between modularizing a system and adding some useful structure into it. Modularity itself doesn't seem to be beneficial, it’s usually the structure that helps

- What's the difference between Badger and GraphNNs with LSTMs?
- We still haven't shown the benefit of multi-agentness
- Modularity - link between Bayesian methods, sparsity, number of params and generalization
- Sparsely distributed systems might have benefits in avoiding catastrophic forgetting
- Modularity also helps in reducing interference
- What is modular, multi-agent-based vs monolithic in the space of NN's vs Badger?
    - There is no clear delineation and understanding of how and where to separate/distinguish those
    - This needs to be understood and defined precisely
- Why did biology choose multi-cellularity?
    - Convenient thing / the right starting conditions?
    - But maybe not the most effective thing if one would design life now
- Meta-reasoning
    - We haven't really touched upon it yet in our research
    - But has links to dreaming and deliberation and economy
    - What pressures are there to make use of meta-reasoning?
        - are there tasks that enforce these pressures?
        - Adaptive Computation Time / variable depth ResNets could have some insights

References/Researchers mentioned during discussion

- Kolchinsky and Rocha, Prediction and Modularity in Dynamical Systems, ECAL 2011
- Hamrick, Jessica B. "Analogues of mental simulation and imagination in deep learning." Current Opinion in Behavioral Sciences 29 (2019): 8-16.
    - there are lots of ways in which animals think that we don't quite do with artificial systems
- Meta-reasoning
    - Jessica Hammrick
- Researchers from neuroscience who consider e.g. temporal pressure in decision-making (e.g. this work on hesitation in CTRNN/Hebbian Learning)
    - Dario Floreano
    - Eiko Matsuda (VTE)
    - Julien Hubert (Memory with spiking)
    - Robert Ward