OpenBrain: Massiveley Asynchronous Neurocomputation

William H. Guss
Machine Learning at Berkeley
2650 Durant Ave, Berkeley CA, 94720
wguss@ml.berkeley.edu

Mike Zhong
Machine Learning at Berkeley
Berkeley CA, 94720
lol@gmail.com

Phillip Kuznetsov
Machine Learning at Berkeley
Berkeley CA, 94720
philkuz@ml.berkeley.edu

Jacky Liang
Machine Learning at Berkeley
Berkeley CA, 94720
jackyliang@berkeley.edu

Instert yourself
Machine Learning at Berkeley
Address
email

We can add footnotes clarifying
Machine Learning at Berkeley
Address
email

equal contributions to the work
(if needed)
Machine Learning at Berkeley
Address
email

Todo list

Abstract

The OpenBrain white paper.

Produce a 4-6 sentence abstract about the paper.

Note: Most of the sections for this document are included as submodules using the following command.

```
\input{section} %imports section.tex
```

The main file should not be modified, except to change the abstract, add new sections, or modify packages!

1 Introduction

Note: We need to motivate substantially the proposal of a drastically different framework of neurocomputation. This motivation should be given according to the latest neuroscience, and a general problem in the field.

Argue against the current state of ML's approach to the problem of AGI by surveying field.

1.1 Intelligence as an Emergence Phenomenon

Establish Conwaynian philosophy on emergent neurocomputation.

Motivate asynchrynous neurocomputation.

2 Asynchronous Neurocomputation

Note: This section will essentially lay out our model minus learning rules. This means we must give theoretical, biological justifications for the algorithm.

2.1 The Core Framework

This section needs an actuall discussion. For now here are the working definitions of Openbrain.

Definition 2.1.1. *A **neuron** $n \in N$ is defined by*

- *a voltage $V_n(t)$,*
- *a decay time τ_n,*
- *a refactory period ρ_n*
- *a voltaic threshold θ_n.*

Definition 2.1.2. *A **connection** $c \in C$ is a tuple $(n_i, n_j, w_{ij}) \in N \times N \times \mathbb{R}$ where n_i is the **anterior neuron**, n_j is the **posterior neuron**, and w_ij is the standard synaptic weight.*

Make connection diagram.

For a neuron n, we denote the set of anterior neurons A_n and the dendritic connections, C_n^a. In the same light we will use the notations P_n and C_n^p to denote the sets of posterior neurons and posterior connections for n respectiveley.

Definition 2.1.3. *A neuron n is said to **fire** if it is not in its refractory period and $V_n(k) > \theta_n$. Then for all $m \in P_n$,*

$$V_m(k+1) \leftarrow V_m(k+1) + w_{nm}\sigma(V_n(k)); \tag{2.1.1}$$

*that is, voltage is propagated to the posterior neurons. Immediately after n fires, it enters a **refractory period** until $k + \frac{\rho_n}{\Delta t}$.*

Definition 2.1.4. *We say that a neuron n experiences **voltage decay** so that for all k,*

$$V_n(k+1) \leftarrow V_n(k+1) + V_n(k)e^{-\Delta t/\tau}. \tag{2.1.2}$$

Combining (2.1.1) and (2.1.2) we get that for a neuron m at time $k+1$

$$V_m(k+1) = V_m(k)e^{-\Delta t/\tau} + \sum_{n \in A'_m} w_{nm}\sigma(V_n(k)) \tag{2.1.3}$$

such that A'_m is the set of anterior neurons which fired at time k.

2.2 Continuous Time Universal Intelligence Measure

In alignment with the philosophy which predicates OpenBrain, we wish to extend the evaluation of our algorithm well beyond its performance in supervised learning tasks; that is, what can be said about the intelligence of OpenBrain as an agent in an environment? A metric more condusive to implementing general intelligence is needed. The answer will motivate an important discussion of representation theory annd learning rules.

2.3 Universal Intelligence

A well established[1, 3, 4] machine intelligence measure is the Universal Intelligence Measure proposed by Legg and Hutter [2]. Drawing from a large amount of disparate literature on the subject they develop a consise definition of the intelligence of an *agent* in an *environment*.

The following exposition may be unnecessary, and we should consider just stating the result of Legg and Hutter.

Environment-agent interaction is defined with respect to an observation space, $\mathcal{O}$, and an action space, $\mathcal{A}$, both of which consist of abstract symbols. The perception space, $\mathcal{P} \subset \mathcal{O} \times \mathbb{R}$, is the combination of observations and rewards.

Definition 2.3.1. *An **environment** μ is a probability measure, specifically defining*

$$\mu(o_k r_k | o_1 r_1 a_1 \dots o_{k-1} r_{k-1} a_{k-1}), \tag{2.3.1}$$

the probability of observing $o_k r_k \in \mathcal{P}$ given a history, a string $o_1 \dots a_{k-1} \in \bigtimes_{i=1}^{k} \mathcal{P} \times \mathcal{A}$.

In the same light the agent definition is given.

Definition 2.3.2. *An **agent** π is a probability measure, giving*

$$\pi(a_k | o_1 r_1 a_1 \dots a_{k-1} o_k r_k) \tag{2.3.2}$$

the probability of the possible action a_k being enacted by π being the environment-agent interaction history.

Include agent environment interaction picture? Would this section be clear given that the reader has explored [2]?

Having defined the basic framework, [2] gives a definition for Universal Intelligence. Let E be the space of all turing complete reward environments, and $K : E \to \mathbb{R}$ be the Kolmogorov complexity of an environment. This complexity is calculated with respect to the length of the string with which a reference machine $\mathcal{U}$ generates the environment.

Definition 2.3.3. *If π is an agent then we say that the **universal intelligence** of π is*

$$\Upsilon(\pi) = \sum_{\mu \in E} 2^{-K(\mu)} V_\mu^\pi \tag{2.3.3}$$

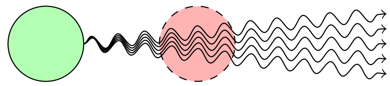
where V_μ^π is the expected reward of the agent in μ,

$$V_\mu^\pi = \mathbb{E}\left(\sum_{i=1}^{\infty} r_i\right) \leqslant 1. \tag{2.3.4}$$

The definition is satisfactory for agents which act synchronously with their environments; that is, the environment waits for the agent to act before giving a new observation. Therefore in Hutter's sense, the framework of [2] describes an agent π embedded in μ.

Despite this, the environments which we normally consider an intelligent agent to 'act well' in are often chaotic and operate with noise which is temporally independent from the agent-environment interaction itself. For example, a real time game does not wait for a player to press a key, and yet the player still receives perceptual information. The intelligence measure proposed fails to encompass agent-environment interactions where the agent has some delay in acting as the environment continues; modeling such delays as $a_k = \varnothing$ is no more enlightening. Why!?

In order to integrate OpenBrain with this framework, we will propose a sort of continuous time universal intelligence measure.



2.4 Universal Approximation and Representation

2.5 (optional) Multiprocess Turing Completeness

If we decide to go down this route, we'll add it's own TEXfile.

3 Conwaynian Learning Rules

4 Experimentation

4.1 Implementation

4.2 Results

5 Conclusion

5.1 Future Work

References

[1] Shane Legg. "Machine super intelligence". PhD thesis. University of Lugano, 2008.

[2] Shane Legg and Marcus Hutter. "Universal intelligence: A definition of machine intelligence". In: *Minds and Machines* 17.4 (2007), pp. 391–444.

[3] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533.

[4] Samuel Rathmanner and Marcus Hutter. "A philosophical treatise of universal induction". In: *Entropy* 13.6 (2011), pp. 1076–1136.

Appendices

A Universal Intelligence Definitions