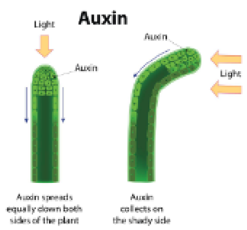
Making distributed training more efficient, enabling the development of computational materials

Summary

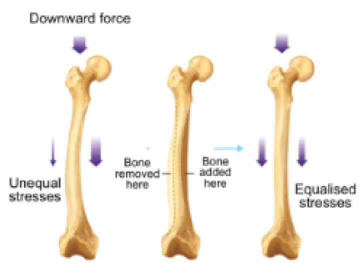
Software behavior is increasingly being “trained” on the basis of large quantities of data, with decreasing need for human engineering. In contrast, there are almost no examples of learned adaptivity in the human-engineered physical world despite countless examples in biology: heliotropism describes how plants alter their shape to position their leaves closer to light. Wolff’s law tells us that calcium is deposited / re-absorbed to / from bones in response to local stresses. An octopus distributes co-localized actuation / computation throughout its body. Cells differentiate and self-organize to produce organs and tissues that perform specialized functions.

All describe phenomena in which collectives of computational units (cells), communicating locally, operate cohesively to achieve an externally defined objective. Designing such systems is challenging: **a) we do not understand algorithmically how to train large scale distributed systems efficiently b) since we do not know the algorithmic principles, we do not know how to imbue the property of “learnability” into engineered systems.**

Heliotropism (light seeking behaviour in plants)



Wolff’s law



Cellular automata / patterning

The ultimate goal of this proposal is to take some initial steps towards **developing physical materials that can be “trained” to perform computations on their inputs and actuate outputs to the real world**. The core technical goal is to demonstrate how to imbue these “materials” with distributed learning algorithms that allow them to learn (or allow training) by demonstration of a series of `[input,output]` examples, where inputs and outputs are physical stimuli and actuation respectively.

Motivation and technical relevance to the Nature Computes Better opportunity space

Distributed training algorithms have the potential to enable compute-in-memory architectures that can be OOM more efficient than Von Neumann architectures by minimizing data transfer. IIUC, **this is a core focus of the NCB opportunity space.**

The principal prerequisite is to demonstrate that distributed learning works at scale. The principal criticism of distributed training algorithms such as Equilibrium Propagation and REINFORCE are that they are high variance (noisy) and so require lots of data to learn. A criticism of compute-in-memory architectures is that reading and writing data is difficult since the storage locations are obscured, such that re-use of learning experience is difficult (the mortal computing problem).

The core research question for this seed call is inspired by the *observation that nature is able to solve these problems*. Depending on their inputs, stem cells differentiate into e.g. skin, neuron or muscle cells and collectively produce functional organs. **(See WS1 below) The goal is to show that we can imbue neurons with priors derived from pre-trained networks that can bias distributed learning in better directions than random exploration**. This is a direct analogue of evolution encoding behavior in DNA and would be important in proving that distributed training need not start "from scratch".

Outputs

<u>Workstream 1 (Algorithmic investigations)</u>

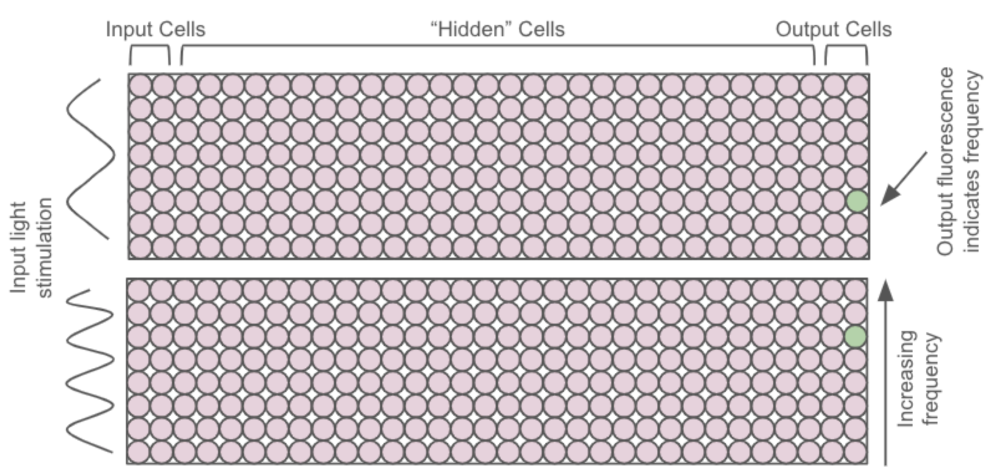
a) Analyze the weights of networks pre-trained with backpropagation on different tasks, fit a distribution to these weights, then use this distribution as a prior for exploration. We will demonstrate that it's possible to design arbitrary networks to perform canonical tasks (e.g. vision, language) from "stem cell"-like artificial neurons, which learn much more efficiently from data by leveraging information from ancestral training runs (i.e. mimicking evolution).

b) We will investigate and improve the performance of these algorithms under non-ideal conditions, such as when they are implemented in vivo / silico by non-ideal components (see **WS2** below).

<u>Workstream 2 (First experiments in a biological system)</u>

The learning rules described in **WS 1** are mechanically simple. The goal of this workstream is to take the first steps towards implementing a minimal version of these learning rules within individual cells, such that collections of cells can be "trained" to perform computation on a set of inputs (delivered to a subset of cells) and actuate some outputs (a distinct subset of cells). **This would be the first demonstration of adaptive (trainable) collective behavior in groups of biological cells.**

Concretely, the goal is to train a community of E. Coli distributed on a 2D plate to "classify" a set of 1D input patterns (see Figure). At one end of the plate, cells will be stimulated by a set of 1D input patterns (e.g. sinusoids with different frequencies). At the other, fluorescence of a layer of output cells will correspond to the "label" (which varies along the y-axis).



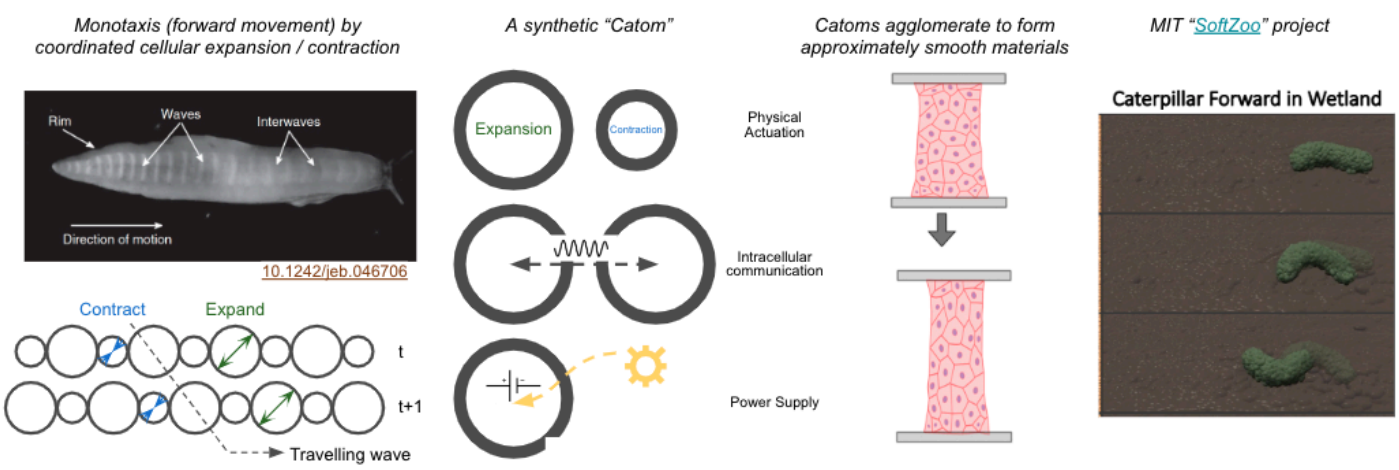
a) Our approach is similar in spirit to what has been implemented in simplified physical circuits. **We propose to compile these learning rules in RNA using Cello**, which relies on the iGEM library of modular genetic "parts" to implement logical functions. This relates to **WS1b**.

a) A second significant challenge is to overcome the non-identifiability of inputs. Bacteria typically communicate via quorum sensing, which involves the exchange of chemical signaling molecules, which may be shared between cells. This project would demonstrate how to overcome these challenges given the constraints of the biological systems - E. Coli can communicate with ~10s of unique signaling molecules, which could be randomized to introduce identifiability. This project would be undertaken in close collaboration with the biological research group.

Why is it of scientific or technological importance?

1. We will demonstrate a "third way" between energy-hungry Von Neumann architectures (which can share training runs across multiple copies) and pure compute-in-memory architectures trained from scratch each time, by identifying common structure across neurons in networks pre-trained on canonical tasks.

2. (Related to "Robotic Dexterity") In robotics there is a hard boundary between computation and actuation. Computation is processed centrally to control external actuators. However, in machine learning terms, everything outside of the central controller defines the "environment" - there is no way for the robot to predict how actuation of a limb will cause sensation by another part of the limb. Contrast this to backpropagation in a neural network - changes can be predicted because gradient information can be propagated through the network. Distributed learning algorithms effectively enable an analogue to the propagation of gradients, allowing relationships to be implicitly learned between different units within a system. **The goal is to allow the physical body of the robot to become a part of that end-to-end differentiable system (Figure 3).**



Why hasn't it been done yet?

Digital systems have historically been hand-engineered (in the Good Old-Fashioned AI sense) but are only very recently becoming learned / adaptive. 2012 was the year that deep learning first worked at scale and 2022 was the year of generative AI for language and images. We are only now at the point at which we can begin to tackle the challenging problem of scaling distributed systems.