Myelin Whitepaper

Jan Nils Ferner, Mathias Fischler, Jeremy Stucki

August 26, 2019

Abstract

Write this after the document is done

Contents

1 Introduction

Second generation neural networks achieve impressive results on all tasks that can collectively be identified as data classification. As of the time of writing, second generation convolutional networks continue dominating every ImageNet classification challenge of the last years [11].

1.1 Limits of Current Approaches to AI

There are still types of tasks that no deep learning topology or technique has been able to tackle so far. Adversarial evaluations of reading comprehension systems indicate that neural networks that excel at evaluating carefully prepared data prove to be extremely fragile when encountering organic "noise" information [8]. The practice of splitting learning into separate training and application steps means that a traditional neural network is unable to continuously learn while working. This way, new information cannot be acquired dynamically as it has to be spoon-fed in the form of carefully prepared test data. Further limitations of current deep learning strategies have been compiled by Gary Marcus [9].

One way to interpret these limitations is as an expression of the growing discrepancy between biological neural networks and the mentioned systems [10].

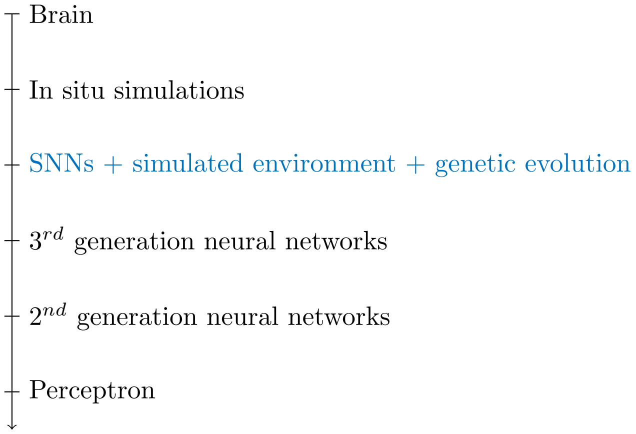
1.2 General Intelligence in Nature

In light of these dissatisfactions with the current state of AI, we take a deliberate step back from current research and ask ourselves what the goal of artificial intelligence research should actually be. In our opinion, AI is not about the automation of simple tasks or the labeling of pictures. Artificial Intelligence should be about intelligence. And the only form of intelligence that can truly stimulate and satisfy the human being is one, that is similar to its own inner workings: A General Intelligence (GI), flexible enough to adapt to an ever-changing environment, enduring enough to continuously learn and change its approach. So far, the only general intelligence that we know of resides within our own heads. It is therefore tempting to build an Artificial General Intelligence (AGI) by simply replicating the human brain in an artificial setting. In practice however, this bottom-up approach proves to be only advantageous when studying in situ micro-models and their effects in isolation [4].

1.3 The Scale of Abstraction

One can imagine all approaches to intelligence as if on a scale of abstraction. One end of this scale, the extreme denominating the absolute lack of any abstraction, is the human brain. The other end has no clear extreme as there cannot be a maximum of abstraction and must be defined arbitrarily. We choose the well-known example of the first-generation perceptron for this task as its shortcomings have been described in great detail by various sources [1]. Our challenge as seekers of General Intelligence is now to analyze various approaches and approximate their position on the scale on order to deduce the abstraction level that will lead to a simulation capable of AGI.

We interpret the limitations of bottom-up brain simulations as a consequence of choosing the wrong level of abstraction, namely one that is too low. In contrast, if we want to place the second-generation approach of Artificial Neural Networks as we train today on the same spectrum of abstraction, they must naturally be placed on the opposite side, as indicated by their reductionist principles. The second-generation systems are too abstract. The optimal level of abstraction necessary for AGI must therefore lie between these two extremes. The solution is neither a chemical copy of the brain, nor does it abstract away the absolute statefulness of the brain as a function of internal state, sensory inputs and time. Third-generation spiking neural networks present themselves as a good compromise between realism and flexibility for conceptual modeling [10]. However, just choosing the appropriate neural model is not enough as we will see.


Brain
In situ simulations
SNNs + simulated environment + genetic evolution
3^{rd} generation neural networks
2^{nd} generation neural networks
Perceptron

2 Environment

2.1 Separation of Mind and Body

In our only role model for General Intelligence, the human brain, a separation of the mind and the body it is residing in is unfeasible [4]. Because we assume that nature is an indicator for the necessary attributes of General Intelligence, the consequent thing to do is not abstract away the bodily environment and its steering needs that gave rise to the mind in the first place [7]. A simulation capable of producing AGI must therefore in our opinion contain not only "brains in jars", but also provide a body with which a simulated organism can interact with other organisms in a social manner and a sandbox-like environment that is susceptible to change. But how far should such a simulation go? Which attributes of the real world are worth simulating in a way that contributes to the emergence of AGI?

2.2 Social Interaction

The human brain does not only thrive on sensory input, it *demands* interaction. Deprived of these stimuli, the brain suffers damage and enforces sensory feedback by employing hallucinations [6]. From this, we can infer that our simulation must provide organisms with information about their surroundings. This includes the ability to communicate with other organisms over some kind of arbitrary protocol. By emitting own messages over the protocol and rewarding organisms that adhere to instructions broadcast in this manner, we can steer how the organisms interpret and shape this common "language".

2.3 Life-Sustaining resources

In order to encourage neuroplasticity and adaptivity, the organisms should be forced out of their comfort zone, as they could stagnate otherwise. To achieve this, the environment continuously alters itself to shift its resources. Another great booster of evolution is competition in a predator-prey environment [3]. Both of these concepts can be described in nature as a consequence of the need for resources used in building survival machines, in Dawkins' words. By introducing the idea of depleting resources in organisms that are necessary for survival, i.e. *energy*, we can guide their behavior by placing objects inside the world which organisms can consume in order to regain energy. The basic energy sources conceived by us are *plants* and *water*, which will both be necessary for the survival of the organism.

The concept of water is simplified for our purposes as a static, lake-like body of water. This resource is generated sparsely at central positions and will not be removed upon consumption. Organisms are forced to interact with each other by the collective need to gather water at the same locations.

Plants are used to naturally control populations. We define them as a resource that is randomly generated many times when the simulation is instantiated. These resources are removed upon consumption. In every tick, the plant has a chance determined by the plant spreading factor ψ to *spread* by spawning a clone of itself at a nearby position. Following these rules, a plant in an unpopulated part of the simulation will eventually spread into a forest. This opens a resource-rich habitat in which organisms can thrive. Organisms can be expected to reproduce until the available plants start to dwindle. At this point, the organisms will starve until either the forest has spread again or all the resources in the area have been consumed, resulting in either migration or extinction. In any case, the number of organisms in the simulation will be bound by the amount of plants that can be consumed. ψ can be tuned according to the hardware limitations faced when running the simulation, as a higher value will result in a lower upper bound on the number of organisms existing at the same time in the world.

The last kind of resource available to an organism is, as an alternative to plants, the consumption of other organisms. This allows for the formation of predator-prey dynamics which, because of the life/dinner principle [3], put selective pressure on the prey to adapt to its environment.

2.4 Dimensionality

Because 3D simulations are much more expensive in terms of complexity and required computational power and none of the characteristics of GI in nature we identified requires a dimension of depth, we choose to run our simulation in a 2D, top-down environment.

3 Training

The driving force behind evolution is the gene seeking immortality [2]. As conventional training with backpropagation proves to be impractical when used with spiking neural networks [10], genetic algorithms seem like a natural training choice when going for a nature-inspired simulation. Additionally, it is our belief that the clear error coefficient used in backpropagation is an unrealistic ideal and, by virtue of being calculated for a specific task, contributes to the aforementioned lack of plasticity in second-generation neural networks.

As we want to train our AI to acquire general skills, the function governing the reproductive success of the organisms for the genetic algorithm has to be general as well. The most general reward system is one that does not reward anything specific at all. The organism that spreads its genes is not determined by a hard-coded reward system, but by the organism's ability to reproduce, just like in real life. This way, we shift the burden of defining what is considered "good fitness" away from a predefined algorithm into the fabric of the simulation itself. For this, we propose giving the organisms the ability to consent to sexual reproduction. This way, the organisms are put under selective pressure by their potential mates by way of sexual selection. [3]

This way of training intentionally does not teach the organisms how to handle specific data. Evolution does not teach you e.g. how to speak English either. It only provides you with the mental capacity to do so.

4 Genetics

As we have discovered in earlier work, combining complex neural networks with genetic algorithms can easily push the computational cost of advancement beyond any reasonable limit [5]. To avoid this pitfall we can use the extra encoding potential provided by the addition of time as an input of our neural network's state [10].

4.1 Genome

Our genome consists of Cluster and Hox Genes. These represent the information which is required to build the neural network.

4.1.1 Important Terms

Cluster Genes Cluster Genes act as a blueprint for a set of neurons and their connections, which we call a cluster.

Hox Genes Hox Genes make up the building plan for the neural network, specifying how clusters should be placed and connected.

Placement Neuron The neuron of the cluster that will be shared with the cluster that it is attached to. Instead of placing a new neuron, a neuron of an already placed cluster is being used as a substitute.

Target Neuron Refers to a neuron on a cluster that has already been placed.

4.2 Example of a genome

As an example, we define three cluster genes and four hox genes.

Naming of Neurons Naming neurons uniquely without creating confusion is difficult. We decided to use the i prefix for cluster definitions (read i_0 as *neuron with the index 0 on the cluster*). Once a cluster is placed, we name the neurons in the order they have been placed in the network. As a prefix, we decided to use n. You can read n_0 as *neuron with the index 0.*

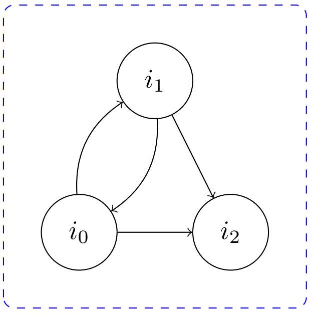


Figure 1: First cluster - The placement neuron is i_0

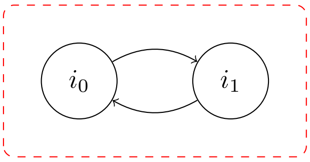


Figure 2: Second cluster - The placement neuron is i_0

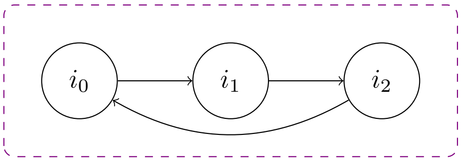


Figure 3: Third cluster - The placement neuron is i_0

The first hox gene places a cluster based on the first cluster gene.

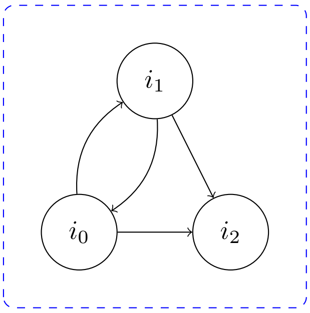


Figure 4: Cluster placed by the first hox gene

The second hox gene places clusters based on the second cluster gene on the clusters placed by the first hox gene. The target neuron is i_2.

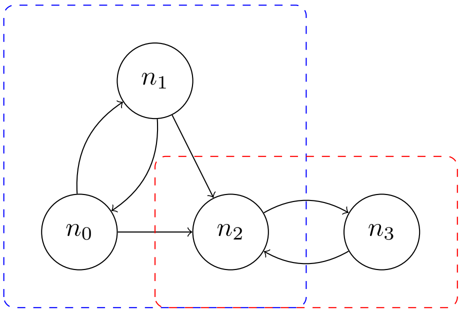


Figure 5: Cluster placed by the second hox gene

The third hox gene places clusters based on the second cluster gene on the clusters placed by the second hox gene. The target neuron is i_1.

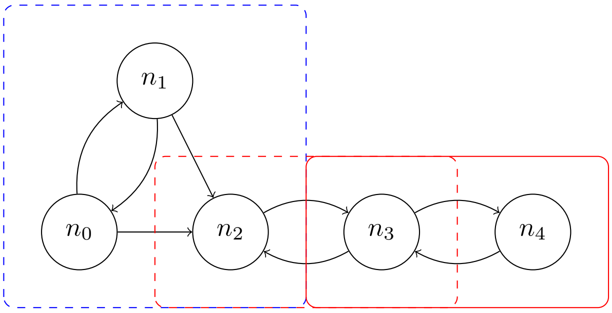


Figure 6: Cluster placed by the third hox gene

The fourth hox gene places clusters based on the third cluster gene on all clusters based on the second cluster gene. The target neuron is i_1.

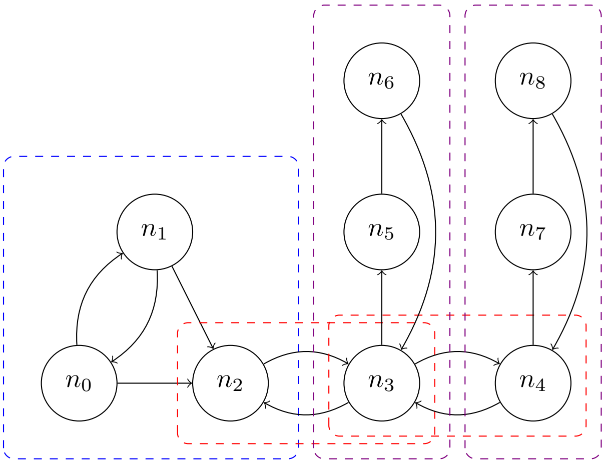


Figure 7: Clusters placed by the fourth hox gene

5 Mutations

5.1 Mutations Overview

1. **Add Neuron**
   Add a neuron to an existing cluster gene. The neuron will be placed on an existing connection. A → B becomes A → C → B, where A and B are existing neurons and C is the newly placed neuron.
2. **Add Connection**
   Add a new connection between a pair of neurons on an existing cluster gene.
3. **Disable Connection**
   Mark an existing connection as disabled.
4. **Nudge Weight**
   Nudge the weight of an existing connection by a small delta value.
5. **Change Placement Neuron**
   Change the neuron that is marked as the placement neuron on a cluster gene.
6. **Add New Cluster**
   Add a new cluster gene and place it using a new hox gene.
7. **Copy Cluster**
   Create a copy of an existing cluster gene and place it using a new hox gene.
8. **Desync Cluster**
   Allow a cluster gene to mutate independently by turning it into a new cluster gene.
9. **Bridge**
   Add a new cluster gene in between two cluster genes that share a neuron.
10. **Add Hox With Existing Cluster**
    Add a new hox gene that places an existing cluster gene.
11. **Change Target Neuron**
    Change the target neuron of a hox gene.
12. **Duplicate Hox**
    Add a new hox gene to the end of the genome with the same configuration as an already existing one.

List of Figures

Bibliography

[1] James A. Anderson. *An Introduction to Neural Networks.* The MIT Press, 1995.

[2] Richard Dawkins. *The Selfish Gene.* Oxford University Press, 1976.

[3] Richard Dawkins. *The Extended Phenotype.* Oxford University Press, 1982.

[4] Yadin Dudai and Kathinka Evers. To simulate or not to simulate: What are the questions? *Neuron*, 84(2):254–261, oct 2014.

[5] Jan Nils Ferner, Mathias Fischler, Jeremy Stucki, and Sara Zarubica. Combining neuro-evolution of augmenting topologies with convolutional neural networks, 2017.

[6] Stuart Grassian. Psychiatric effects of solitary confinement. *Journal of Law and Policy*, 22, 2006.

[7] G. Jekely. Origin and early evolution of neural circuits for the control of ciliary locomotion. *Proceedings of the Royal Society B: Biological Sciences*, 278(1707):914–922, dec 2010.

[8] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328, 2017.

[9] Gary Marcus. Deep learning: A critical appraisal. *CoRR*, abs/1801.00631, 2018.

[10] Hélène Paugam-Moisy and Sander Bohte. *Computing with Spiking Neuron Networks*, pages 335–376. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.