Exploiting Developmental Plasticity in the Evolution of Deep Learning Architectures

Introduction. Deep learning has provided computational means to address difficult problems such as image classification and automated language translation. Deep learning uses large quantities of data to train the connection weights of artificial neural networks (ANNs). Unfortunately, the extensive machine learning expertise required to design model architectures (the pattern of connectivity between artificial neurons) has bottlenecked rollout of deep learning to commercial and industrial applications. Researchers at Google Brain [1] and Sentient Technologies [2] have successfully used **evolutionary algorithms to automatically design deep learning architectures**, reporting state-of-the-art performance. Through repeated evaluation, selection, and recombination of a population of candidate architectures, this technique evolves architectures that can be trained effectively for particular tasks.

In existing work, plasticity (changes to an ANN in response to training data) is limited to connection weights; network architecture is adjusted only by mutation. In contrast, biological neural subnetworks develop in a manner responsive to environmental stimulus and the activity of other subnetworks. The theory of **neuronal group selection (NGS)** posits that development generates an excess of neural subnetworks, of which only those that successfully integrate into overall brain functionality are retained [3]. This paradigm is hypothesized to **enhance plastic adaptation to environmental conditions and enhance robustness to mutation** [4]. In deep learning models, node and connection removal techniques mirroring NGS have been used to shrink parameter count by an entire order of magnitude without incurring accuracy loss [5].

Large-scale artificial neuroevolution relies on indirect genetic encoding schemes that define an algorithm that constructs an ANN instead of directly specifying individual connections. Indirect encodings generate ANNs with recurring structures and symmetries, which are useful in many problem domains. However, indirect-encoded ANN performance can often be improved through irregular refinements that were inhibited by the constraints of the indirect encoding [6]. By pruning away nodes based on network activity patterns, **NGS provides a potential avenue for such irregular refinement**.

By increasing robustness to mutation, enabling irregular refinement, pruning model parameters, and permitting architectural plasticity in response to the training process, exploiting NGS in evolutionary algorithms for deep learning architectures has the potential to **increase the effectiveness, compactness, and versatility of generated architectures while reducing computational cost**.

Implementation. Under the NGS regime, candidate solutions will consist of two components: (1) an initial architecture A, and (2) a vector v of network pruning parameters. These components will be encoded and recombined separately using standard neuroevolution and genetic algorithm techniques, respectively. To evaluate a candidate solution, the connection weights of the initial architecture A will be trained for a fixed number of iterations. Then, a pruning decision will be made for each network node by comparing a cutoff value (defined by v) to a quadratic weighting (also defined by v) of the following metrics: (a) node activation, (b) correlation of node activation with activation of succeeding nodes, (c) correlation of node activation with loss, and (d) backpropagation gradient through the node. Following pruning, training will resume for a fixed number of iterations. Candidate solution fitness will be assessed as the validation set performance of the final trained network. Figure 1 compares NGS with existing methods.

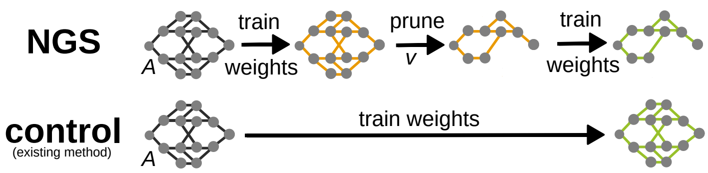

Figure 1: A schematic comparison of a candidate NGS solution (top) with a candidate solution under existing methods (bottom).

Research Plan. First, I will conduct **exploratory experiments to assess the capacity of NGS to promote irregular refinement, evolvability, adaptive plasticity, and parsimony** at the Michigan State University High Performance Computing Center. For these experiments, I will use HyperNEAT, a standard indirect encoding for artificial neuroevolution [6]. These experiments will assess four variants of the NGS algorithm with (a) single or multiple pruning passes and (b) pruning performed on nodes or connections. I will evolve ANN architectures for a variant of the bit mirroring problem, which is designed to enable experimental manipulation of symmetry in the problem domain [6]. I will assess the capacity of NGS to enable irregular refinement by comparing performance of NGS-HyperNEAT variants and HyperNEAT (control) across a spectrum of problem regularities. Greater relative performance of NGS-HyperNEAT variants at low problem regularity would confirm that NGS enables irregular refinement. I will assess the effect of NGS on evolvability by comparing between NGS and control treatments the phenotypic novelty and viability of mutant offspring of evolved architectures [7]. To assess adaptive plasticity, I will use NGS and control methods to evolve architectures to handle one set of bit mirroring symmetries and then test performance on a bit mirroring problem with different symmetries. Finally, I will compare the effect of parsimony selection pressure on performance and network size between NGS and control treatments. I expect NGS to boost parsimony, irregular refinement, evolvability, and adaptive plasticity. Comparing these metrics across four NGS variants will help reveal the most effective formulation of the NGS algorithm.

Next, I will **assess the performance of NGS-enabled architecture evolution on benchmark deep learning datasets**. I will leverage the results of my exploratory experiments to **seek collaborators in industry** for these benchmarking experiments through preexisting connections between my research group and Sentient Technologies. These larger-scale experiments will employ the CoDeepNEAT encoding [2], a layer-based indirect encoding developed specially to evolve deep learning architectures. I will benchmark NGS-CoDeepNEAT against existing results for CoDeepNEAT on datasets for computer vision (CIFAR-10/100) and language modeling (PTB). I expect performance to meet or exceed state-of-the-art.

Intellectual Merit. My research will introduce a **novel bio-inspired machine learning technique**, paving the way for continuing advances in bio-AI. In addition, my experimental results describing the relationship between developmental plasticity and evolvability will **directly inform debate among evolutionary biologists** over the controversial Extended Evolutionary Synthesis [8]. I will use the Modular Agent Based Evolution Framework [9] for my exploratory experiments to enable **easy plug-and-play reuse of software components by other researchers** for further experiments investigating the interplay between plasticity and evolution.

Broader Impacts. I aim to **make powerful deep learning methods accessible to non-experts** by using NGS to reduce the computational cost of evolving deep learning architectures. Partnering with industry will enable my work to help solve real-world problems by removing barriers to the development of new AI products and services.

I plan to **work with undergraduate researchers to demonstrate applications of deep learning architectures evolved with NGS** in natural language processing, computer vision, and game-playing agents and **create interactive web media to showcase their work to the general public**. The appealing, concrete nature of these applications and **funding through NSF BEACON**, which places dozens of underrepresented students into evolution-focused research laboratories every summer, will help us recruit from a broad spectrum of students.

References. [1] Real *et al.* LS Evo of IC. *arXiv preprint* (2017), [2] Miikkulainen *et al.* Evo DNNs. *arXiv preprint* (2017), [4] Downing. *Intel Emerg* (2015), [3] Sanes *et al. D of Nerv Sys* (2011), [5] Song *et al.* in *NIPS 28* (2015), [6] Clune *et al. IEEE T EC* (2011), [8] Laland *et al. Nature* (2014), [9] Hintze & Bohm. *MABE* 2017