Literature Review Plan

A literature review is a written summary of journal articles, books and other documents that describes the past and current state of information, organizes the literature into topics and documents a need for a proposed study. It can be seen as the presentation, classification and evaluation of what other researchers have written on particular subject.

# Plan

1. What is Neuroevolution
2. When did neuroevolution start becoming a thing
3. Why neuroevolution what do we use it for
4. How is nueroevolution implemented
5. How is nueroevolution more beneficial than others in its field how does it compare?
6. What are its disadvantages?
7. What is its potential?
8. Whats the future of neuroevolution?

Proposal (kind of)

Artificial neural networks are applied to many real-world problems, ranging from pattern classification to robot control. In order to design a neural network for a particular task, the choice of an architecture (including the choice of a neuron model), and the choice of a learning algorithm have to be addressed. Evolutionary search methods can provide an automatic solution to these problems. New insights in both neuroscience and evolutionary biology have led to the development of increasingly powerful neuroevolution techniques over the last decade. This paper gives an overview of the most prominent methods for evolving artificial neural networks with a special focus on recent advances in the synthesis of learning architectures.

What is Neuroevolution?

* The general idea of Neuroevolution is Darwinian principles being applied in computer algorithms and artificial neural networks combined with this is and deep learning and neurocomputation is neuroevolution. The intersection of neural networks and darwinan principles being applied in computer algorithms is neuroevolution.
* Neuroevolution is a subdivision of Artificial Intelligence that looks at employing evolutionary algorithms to determine the architecture of neural networks. The evolutionary algorithms are used to define the topology of the artificial neural network.
* Evolutionary computing uses the same characteristics as biological evolution to search for optimal solutions to computational problems. Neuroevolution is a method to train neural networks by using this problem-solving technique. The goal of this paper is to present an overview of all the neuro-evolution techniques and to give a summary of this promising training method. (An overview of neuroevolution techniques)
* **What is an artificial neural network topology?**
* **What is an evolutionary algorithm or evolutionary computing methods?**
  + In [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence), an **evolutionary algorithm** (**EA**) is a [subset](https://en.wikipedia.org/wiki/Subset) of [evolutionary computation](https://en.wikipedia.org/wiki/Evolutionary_computation),[[1]](https://en.wikipedia.org/wiki/Evolutionary_algorithm#cite_note-EVOALG-1) a generic population-based [metaheuristic](https://en.wikipedia.org/wiki/Metaheuristic) [optimization](https://en.wikipedia.org/wiki/Optimization_(mathematics)) [algorithm](https://en.wikipedia.org/wiki/Algorithm). An EA uses mechanisms inspired by [biological evolution](https://en.wikipedia.org/wiki/Biological_evolution), such as [reproduction](https://en.wikipedia.org/wiki/Reproduction), [mutation](https://en.wikipedia.org/wiki/Mutation), [recombination](https://en.wikipedia.org/wiki/Genetic_recombination), and [selection](https://en.wikipedia.org/wiki/Natural_selection). [Candidate solutions](https://en.wikipedia.org/wiki/Candidate_solution) to the [optimization problem](https://en.wikipedia.org/wiki/Optimization_problem) play the role of individuals in a population, and the [fitness function](https://en.wikipedia.org/wiki/Fitness_function) determines the quality of the solutions (see also [loss function](https://en.wikipedia.org/wiki/Loss_function)). [Evolution](https://en.wikipedia.org/wiki/Evolution) of the population then takes place after the repeated application of the above operators.
  + Evolutionary algorithms are search algoirthms gleaned from organic evolution. These algorithms were developd more than 30 years ago when researchers were trginy to come up with ideas to solve problems to imitate the intelligent capabilities of individual brainds and populations. The former approach, emphasizaing an indiviuals intelligence, led to the developement of research topics such as ANN and knowledge based symbolic artificial intelligence. Modelling organic evolutioj provides the basis for a variety of concepts such as genotype, genetic code, phenotype, self adapatation etc. all of these concepts were incorporated into evolutionary algorithms. Nature is the inspiration. Biomimicry and them tings there. (Evolutionary Alogirthms in Theory and Practice, Evolution Strategies)
  + This is how evolutionary algorithms work in principle: They model the collective learning process within a population of individuals, each of which represents a search point in the space of potentiaial solutions to the given problem. The starting population is initialised by an algorithm dependent method and evolves towards successively better regions in the search space by means of crossover, mutation and selection. The environ,ent delviers a quality information(fitness value) for new search points and the selection process favours those individuals of higher quality to reproduce more often than worse indivuals. The recombination allows for mixing of parental information while passing it to their descendants and mutation introduces innovation to the population. In an evolutionary framework the fitness of an individual is measured by its proprensity to survive and reproduce in a particular environment.
  + Pseudo code: Step One: Generate the initial [population](https://en.wikipedia.org/wiki/Population) of [individuals](https://en.wikipedia.org/wiki/Individual) randomly. (First generation) Step Two: Evaluate the [fitness](https://en.wikipedia.org/wiki/Fitness_function) of each individual in that population (time limit, sufficient fitness achieved, etc.) Step Three: Repeat the following regenerational steps until termination: Select the best-fit individuals for [reproduction](https://en.wikipedia.org/wiki/Reproduce). (Parents) [Breed](https://en.wikipedia.org/wiki/Breed) new individuals through [crossover](https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)) and [mutation](https://en.wikipedia.org/wiki/Mutation_(genetic_algorithm)) operations to give birth to [offspring](https://en.wikipedia.org/wiki/Offspring). Evaluate the individual fitness of new individuals. Replace least-fit population with new individuals.
* **How do neural network topology and evolutionary algorithms work to form neuroevolution?**
  + **Neuroevolution uses evolutionary algorithms to develop artificial neural networks, with meta optimal parameters, topologies and rules. It uses the mutating and selecting concepts of evolutionary algorithms to select the best performing neural networks.** neuroevolution seeks to develop the means of evolving neural networks through evolutionary algorithms.

**History is Neuroevolution? – (https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning)**

* The first neuroevolution algorithm appeared in the 1980s. At the time, its small group of practitioners thought it might be an alternative to the more conventional ANN training algorithm called backpropagation (a form of stochastic gradient descent).
* Neuroevolution has been developed over the last couple of decades, it is in- teresting to know which types of algorithms are used more frequently and in which direction the field is heading. The first representations had a fixed topology and a fixed number of hidden nodes. These representations had some drawbacks and other representations were developed to circumvent these disadvantages. These methods ranged from direct representations, like NEAT, to developmental and indirect representations. Recently how- ever, fixed topology networks like CMA-evolution strategies and CoSyNE are gaining popularity. This might be because these simpler representations do have a smaller search space, because they do not have to search for the right topology. (An overview of neuroevolution techniques)
* Figure 2.6 shows a timeline from the first application of neuroevolution until now. NEAT is still the most popular representation to date, followed by CMA-ES. Developmental approaches have not seen much use in recent years and AGE is by far the most succesful implicit representa- tion. AGE, CoSyNe, and CMA-ES have all had similar or better results than NEAT at several benchmark tests, the popularity of the NEAT algorithm could be ascribed to its understandability combined with its robustness.

**Why neuroevolution what do we use it for**

* Reinforcement learning.
* Neuroevolution is most succesful in reinforcement learning problems where controllers need to be evolved. especially when considering controlling an object that needs to stay balanced, a helicopter or pole, or controlling non- player characters in computer games or board games. Neuroevolution has been succesful in classification tasks at some occasions, but is not clear what the true potential is for neuroevolution algorithms in this area.
* In general, neuroevolution is more suitable for unsupervised learning problemes. Unsupervised learning can be described as the area of machine learning where a training set of of input-output pairs does not exist (An overview of neuroevolution techniques)
* There are popular benchmark problems and some of the other problems to which neuroevolution has been succesfully applied. (An overview of neuroevolution techniques)
* balancing two poles on one card that have a different length and balancing poles without knowing all the information available. On these tasks neuroevolution outperforms other reinforcement learning techniques
* While neuroevolution techniques are mostly tested with the help of bench- mark problems, there are multiple examples of other applications where neu- roevolution performs well. a promising field is perhaps the evolution of agents in computer games. Schrum and Miikkulainen, [SM08], evolved opponents in computergames that were able to distinguish between objectives of different importance in a simple two dimensional game.
* In [AbBR01], Aharonov-Barki et. al. evolved a controller for an agent that used memory to search for food, and Bryant and Miikkulainen, in 2003, [BM03], set up an experiment where a team of identical agents could work together and perform different tasks to achieve a team objective. While these experiments involve games that are much simpler than the average game that is played nowadays, the need for intelligent behavior of *non-player characters* in computer games is increas- ing and reinforcement learning techniques like neuroevolution could provide this behavior.
* Bryant and Miikkulainen performed another interesting ex- periment in 2007, [BM07] , where the goal was to develop visibly intelligent behavior in NPC’s. They argued that good behavior for an NPC does not necessarily need to be optimal behavior, but behavior that mimicks human behavior. They conducted an experiment and concluded that a form of neu- roevolution performed best when non-player characters had to mimick human behavior.
* Neuroevolution has also been applied to evolve systems that can play spe- cific games. [CF99] used neuroevolution to evolve a controller that could play checkers and that could beat a master player on several occasions. Kaikhah and Garlick, [KG00], used their version of the ESP algorithm to evolve a blackjack player that used card counting strategies to gain an advance of the house. Neurevolution has also been used as a succesful controller in vehicle simulations for a car, [CLL10], and a helicopter, [KW09].
* Neuroevolution can also be used for classification tasks, classification is a supervised learning task because the classes are known in advance. Evo- lutionary algorithms are not specifically suited for these tasks. Johan Hägg,
* (**https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning**)Underneath the hood of deep learning which is responsible for so much distruptive and innovative technologies, such as … , is the latest form of Artificial Neural Networks (ANN). ANN is an attempt to simulate a collection of neuron like componenets that send signalas to each other. That is the underlying mechanism behind deep networks in deep learning. Where do these connections come from? Our brains is year of evolution so we want to do the same for the neural networks.­ The 100-trillion-connection architecture of our human brain evolved through a Darwinian process over many millions of years. (use a reference here). In short, the brain – including its architecture and how it learns, is a product of natural evolution, and neuroevolution can probe all the factors that contribute to its emergence.
* It a way of opitmiatisation as opposed to other popular optimisation techniques such as gradient descent and more.

**How is it being implemented, what are people doing? – (https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning)**

* Kenneth stanely, NEAT, HyperNEAT, novelty search
* The NeuroEvolution of Augmenting Topologies (NEAT) algorithm is the most widely used algorithm at this mo- ment, although both CMA-ES and CoSyNE have shown better results at certain benchmark tests. The popularity of NEAT could be ascribed to the fact that it is more robust, because the amount of hidden nodes is not chosen in advance. (An overview of neuroevolution techniques)
* How do you evolve an artificial brain to solve a problem? Example if you want to evolve a neural network to control a robot to walk, we will have a robot body in a physics simulator, some ANNs and because we don’t know how to solve the task, we generate a population of 100 random ANNs. ~~Thus a fixed topology of ANNs, the weights of the predetermined architecture would be randomized in each of the 100 indivuals in the population.~~
* The application of neuroevolution methods to RL is not a new phenomenon ([18], [19], [20]), but they were only recently shown to be competitive with the Deep RL approach in [21],

and [16]. In the aforementioned papers, evolutionary methods were used, Natural Evolution Strategies in [21], and a Genetic Algorithm in [16], to determine the parameters of neural networks to play Atari games, and for agent locomotion in a simulated physical environment.In both papers, these approaches yield results that are comparable to Deep RL, and even better in a few cases. One advantage is that evolutionary methods are highly parallelizable, since evaluation does not need to be carried out in a sequential fashion, meaning they do provide a significant speedup when it comes to learning. Unfortunately, this parallelization requires access to a lot of computational power to realize this advantage. For example, in [21] the task of 3D humanoid locomotion was evaluated at different numbers of CPU cores, and reaching a predetermined score takes 11 hours using 18 cores, and 10 minutes using 1440 cores. The speedup is significant but quite costly.

**How is neuroevolution more beneficial than others in its industry?**

**What are its disadvantages? – (**[**https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning)**](https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning))

Over the decades since the first fixed-topology neuroevolution algorithms began to appear, researchers have continually run into the frustrating reality that even as the algorithms create new possibilities, the brains they can evolve remain far from what evolved in nature. There are many reasons for this gap, (maybe give some reasons??) but a fascinating aspect of the field is that every so often a surprising new insight into the workings of natural evolution emerges, resulting in a leap in the capability of neuroevolution algorithms. These results highlight the complexity of my God, and the ‘mysteriousness’ (maybe use another word?) of nature.

What does it mean to make progress in neuroevolution? In general, it involves recognizing a limitation on the complexity of the ANNs that can evolve and then introducing an approach to overcoming that limitation.