Evolving Plastic Neural Networks for Online Learning: Review and Future Directions

Oliver J. Coleman and Alan D. Blair

School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
{ocoleman,blair}@cse.unsw.edu.au

Abstract. Recent years have seen a resurgence of interest in evolving plastic neural networks for online learning. These approaches have an intrinsic appeal – since, to date, the only working example of general intelligence is the human brain, which has developed through evolution, and exhibits a great capacity to adapt to unfamiliar environments. In this paper we review prior work in this area – including problem domains and tasks, fitness functions, synaptic plasticity models and neural network encoding schemes. We conclude with a discussion of current findings and promising future directions, including incorporation of functional properties observed in biological neural networks which appear to play a role in learning processes, and addressing the "general" in general intelligence by the introduction of previously unseen tasks during the evolution process.

Keywords: plastic neural networks, evolution, online learning, meta-learning.

1 Introduction

Recent years have seen a resurgence of interest in evolving plastic neural networks for online learning. These approaches have an intrinsic appeal – since, to date, the only working example of general intelligence is the human brain, which has developed through evolution, and exhibits a great capacity to adapt to unfamiliar environments. In this paper we review prior work in this area – including problem domains and tasks, fitness functions, synaptic plasticity models and neural network encoding schemes. We conclude with a discussion of current findings and possible future directions.

In this review, a *plastic neural network* is one in which the strengths of synapses may change during the networks operational life. *Online learning* refers to the ability of an agent to discover or learn about some property of its environment, typically by exploration, which it has not been exposed to previously or which changes during its life time, and then exploit this knowledge in order to achieve a goal. *Online adaptation* refers to a robustness or ability to adapt to internal or external perturbations or changes in input and output ranges; in other words, the ability to maintain homeostasis. These latter two definitions are very similar to terms defined by Mouret and Tonelli [19]: an agent is said to possess *behavioural robustness*, which is akin to online adaptation, when it can maintain the same qualitative behaviour despite environmental or morphological changes; and an agent is said to exhibit *behavioural change*, which is akin to online learning, when in a reward-based environment a change in reward causes it to

M. Thielscher and D. Zhang (Eds.): AI 2012, LNCS 7691, pp. 326–337, 2012.

adopt qualitatively new behaviours until a new optimal behaviour has been found. While this review describes progress in evolving plastic neural networks for online learning, work on evolving plastic neural networks for online adaptation has enough similarities to provide useful insights. We include work on online adaptation in the sections on synaptic plasticity, scaling/competition and neuronal excitability regulation. Finally, it should be noted that "adaptive network" is not synonymous with "plastic network": a network may be adaptive without having plastic synapses [11,18,25].

2 Problem Domains, Tasks and Fitness Functions

In the evolution of neural networks for online learning, evaluating the fitness of a candidate requires testing the ability of the neural network to learn something about its environment and to then exploit that knowledge to achieve one or more goals. This requires producing environments that are different in some way for each generation and/or change in some way during a fitness evaluation [4,20,25,27].

Initial work on online learning, carried out in the early 1990s, focused on a supervised learning paradigm [3,4,13]. Fitness evaluation typically took the form of a training phase during which sets of exemplars are presented to the network, and then an evaluation phase where the network is presented all the input vectors from the training set and the fitness is the percentage of correct corresponding output vectors produced by the network [3,4]. These studies were largely proof-of-concept in nature and used simple network models and simple, for example single-bit or linearly-separable, binary training examples. The main limitation with this method is that the generalisation performance of the evolved learning algorithms is not tested (not to be confused with the ability to learn all the members of any particular set of exemplars).

Research on evolving neural networks for online learning re-emerged in the early 2000's – this time focusing on reinforcement learning domains. In most reinforcement learning experiments the fitness of an agent has been the amount of reward it received during its lifetime [20,25,27] or is at least strongly correlated with the reward received [15,23]. Use of multiple reward signals, corresponding to multiple goals, has not been explored.

Most tasks studied to date have been designed to provide a simple reinforcement learning environment in order to evaluate the ability of an agent to perform learning. They have required learning a simple association that changes during the agents life time, such as which type of flower provides the most reward in a simple bee foraging task [20, 21, 26], which object types are food or poison in a slightly more complex foraging task [27], or which arm of a T-maze contains a high-value reward [21,22,25]. Variations to increase the difficultly of the T-maze task include use of a double T-maze, which has four arms instead of two, or requiring the agent to learn a possibly non-linear association between the perception of objects and their reward value [22]. Risi and Stanley [23] studied a significantly more difficult version of the T-maze task in wihch the environment was continuous rather than discrete and the controller takes as input only raw output from 5 rangefinder sensors (as well as a reward signal) rather than signals from special sensors indicating arrival at key locations.

Tonelli and Mouret [28, 29] studied a purely associative task, similar to those using a supervised learning paradigm, in which an agent is required to learn associations between each possible input and output pattern. The input and output were vectors of binary values, with a reward signal added to the input. To simplify the task only one bit of the input and output vector each has value 1 while all others have value 0. The difficulty of the task was increased by increasing the size of the vectors. This problem domain is interesting because the number of associations that must be learned can be large, while other problem domains studied have required only learning one or a few associations, and also because it does not require simulation of an environment or learning fixed behaviours for invariant properties of the environment (other than the ability to determine which output to set high for which input, of course). A much more complex task in this problem domain would be to allow arbitrary input and output patterns, instead of setting only one bit high per pattern.

Khan and Miller [15] introduced a modified version of Wumpus world, where pits and the Wumpus only harm rather than kill the agent, as well as a competitive version of Wumpus world where the Wumpus acts as a predator. Rather than a reinforcement signal being provided, the agent can perceive its "energy" level, which is affected by environmental factors and achieving goals.

It has been noted by several authors that most tasks requiring online learning also require some fixed behaviours, and that often the fixed behaviours can be evolved much more easily than the learning behaviours [20,21]. Indeed, if the goal is to test the ability of an agent to perform online learning, then the experimenter must take care to ensure that the task cannot be solved with purely fixed behaviours [3]. Preventing the evolutionary process from becoming stuck on the local optima of fixed behaviours has typically been addressed by employing specially designed environments and fitness evaluation functions that minimise the advantage of purely fixed behaviours and that strongly favour learning behaviour [3, 21, 23, 29]. In order for the approach of evolving neural networks for online learning to be practical for a wider range of problem domains – where a large number of behaviours may need to be fixed for a successful agent – the minimisation of this requirement represents a significant future challenge.

One novel approach to alleviating the problem of local optima in evolutionary algorithms is novelty search. Novelty search replaces the objective or goal-based fitness function in an evolutionary algorithm with a function that measures the behavioural novelty of an individual with respect to the behaviours exhibited by other individuals in the current and all preceding generations. The main idea behind novelty search is that behaviours that are different in interesting ways from other behaviours collapse to the same objective-based fitness, so objective-based search essentially ignores them even though they may be pathways to a solution. This method has shown promising results in evolving neural networks for online learning [21], where it was able to evolve solutions more quickly than objective-based search on two common benchmark tasks. It has also been used successfully in conjunction with a traditional objective-based fitness function using a multi-objective evolutionary algorithm [29]. However, some studies suggest that novelty search may not scale well to more complex problems [6] and that in some cases it is just as hard to find a good novelty metric as it is to find a good objective-based fitness function [16].

3 Synaptic Plasticity Models

The most common approach to synaptic plasticity in evolved neural networks thus far has been to define either one or four plasticity rules. These may be applied to some or all connections in the network. In the case of a single rule type, this rule has either been evolved [3] or fixed [21]. Floreano and Mondada introduced a set of four rules in 1996 which were then adopted by several subsequent authors [14, 18], with the genome encoding which of the rules is to be used by each connection. From Floreano and Mondada, 1996 [10, pg. 3]: *"The four types of synaptic change are as follows ... The simplest learning mechanism is plain Hebb, whereby synapses can only be strengthened ... The postsynaptic rule is similar to the plain Hebbian rule, but also decreases the synaptic efficacy when the postsynaptic unit is active and the presynaptic unit is not ... in the presynaptic learning rule the decrement takes place when the presynaptic unit is active, but the postsynaptic unit is inactive ... The covariance rule here takes the form of a synchronous-activation detector: if the presynaptic and postsynaptic activity levels differ by more than half the maximum node activation, the synaptic efficacy is reduced in proportion to that difference, otherwise it is increased in proportion to the difference."*

The second most common approach is to use a parameterised weight update rule where the parameters are evolved either for a globally applied rule [25, 26, 30], for a fixed number of rules [27], for each modular grouping of neurons [20], or per connection [8, 9, 22, 23]. No work has compared the effects of these different levels of granularity.

One of the first parameterised rules was introduced by Niv et al. in 2002 [20] and has been used by several subsequent authors [22, 23, 25, 26, 28, 29]. It consists of a correlation term, pre- and post-synaptic terms and a constant term for heterosynaptic updates, with evolved coefficients for each term:

$$\Delta w = \eta(Axy + Bx + Cy + D) \tag{1}$$

where η is the learning rate and x and y are the pre- and post-synaptic activation.

Di Paolo et al. [8, 9] studied spike-timing dependent plasticity (STDP) models. The weight update rule was asymmetric such that if a pre-synaptic spike occurred before a post-synaptic spike then the weight is increased (potentiated), and conversely if a pre-synaptic spike occurred after a post-synaptic spike then the weight is decreased (depressed). Parameters for the time-window and amount of change were evolved per connection.

Di Paolo [8] performed a direct comparison between rate-based and spiking models. It is noted that a direct and fair comparison is not simple; efforts were made to make the comparison as fair as possible, for example by modifying the weight update rule such that the initial direction of weight change in the rate-based model is unbiased by the initial random activation values of neurons, as it is in the spiking model. Evolved rate-based networks had significantly lower performance than spiking networks in the photo-taxis task studied. This was due to the rate-based networks being much slower to converge to the required weight values: *"STDP controllers are rapidly able to define a direction of weight change depending on the relation between the plastic rules and the* *neural properties ... whereas rate-based plastic controllers take much longer to settle into a given range.*" The practical significance of this result could be questioned: it would be simple to encode initial weight values to overcome this problem. However, in some cases (perhaps developmental network models), such encoding of initial weights may not be possible, thus making the rapid convergence of the spiking networks desirable. Additionally, the STDP rule employed was found to produce more stable networks even in the presence of noise introduced into neuronal activation.

The above approaches all use some kind of relatively fixed rule or rules to change synaptic efficacy, albeit with some evolvable parameters. In contrast there have been two proposed approaches to evolving arbitrary plasticity rules. Khan and Miller [15] evolved genetic programs that controlled changes to synaptic efficacy, as well as several other functional aspects of neurons. While this work is intriguing, and the model developed was demonstrated to be able to evolve solutions to complex tasks, it is not clear to what extent evolving arbitrary rules contributed to the results, positively or negatively. The model developed was very complex and a significant departure from anything that had gone before. The tasks chosen were different from any others used in similar work. The rules that evolved have not been analysed in any studies to date.

Risi and Stanley [22] also developed a novel synaptic plasticity system able to evolve arbitrarily complex rules. They performed a comparison between it and two parameterised rules similar to that introduced by Niv et al. [20] (described above). These models employed the HyperNEAT encoding scheme (see Section 5). The three models of synaptic plasticity compared were:

Plain Hebbian: the update rule is

$$\Delta w = \eta xy \tag{2}$$

where η is the learning rate. The genome function has an output for a synapses existence and for the plasticity learning rate if it exists.

Hebbian ABC: the update rule is

$$\Delta w = \eta(Axy + Bx + Cy) \tag{3}$$

The genome function has an output for each of a synapses existence, the learning rate and the coefficients A, B and C.

Iterated: in this model the genome function is queried throughout the networks life time for each synaptic weight value, instead of only for the initial synapse parameters, and has three inputs additional to the coordinates of the pre- and post-synaptic neurons: the pre- and post-synaptic activation and the current weight value. Thus instead of a fixed Hebbian-type rule, arbitrarily complex weight update rules can be evolved based on location, activation levels and current weight value.

Risi and Stanley tested the networks on a double T-maze task where the complexity of the task could be increased by introducing a non-linearly separable perception and reward value mapping. Evolution using the Plain Hebbian rule could not find solutions for the task at any complexity level, evolution using the Hebbian ABC and Iterated models could find optimal solutions for the linearly separable reward signals task, but only by using the Iterated model could optimal solutions be found for the non-linearly separable reward signals task. For the higher complexity task an optimal solution required a non-linear learning rule which could not be encoded by the Hebbian ABC model. They conclude that although a different network topology may have allowed a less general plasticity rule to solve the non-linear problem in this instance, it is nevertheless risky to make a priori assumptions about which plasticity rules will be suitable. The rules that were evolved with the Iterated model were not analysed.

Both the evolved genetic program and evolved function plasticity rules developed by Khan and Miller and Risi and Stanley, respectively, required significantly more computation time than models using fixed rules, demonstrating a trade-off between computational requirements and generality. While Risi and Stanley demonstrate that in some cases more generality may be required, it is not clear (nor necessarily suggested by the authors) that the level of generality provided by these systems is necessary, or desirable, due to the computational considerations and increased search space in an evolutionary context. Indeed, Risi and Stanley found that the more general model took much longer to find nearly optimal solutions than a parameterised fixed rule model.

Most studies have used either relatively fixed rules, for example the set of four fixed rules popularised by Floreano and Mondada [10] and the parameterised rule introduced by Niv et al. [20], or arbitrarily flexible rules such as the genetic programs introduced by Khan and Miller [15] and the evolved functions studied by Risi and Stanley [22]. While the study performed by Risi and Stanley directly compares simple parameterised rules with arbitrarily flexible rules, a possibly interesting research direction could be to compare plasticity rules that fall between these two extremes. This could especially be the case in plasticity models such as STDP, where evolved parameters could control aspects such as asymmetry, timing-direction, time-frames, plasticity modulation dependent on pre- and post-synaptic firing rates, etc.

Another interesting question is what sorts of synaptic plasticity rules were evolved by the models allowing arbitrary complexity. As noted, the studies performed by Khan and Miller, and Risi and Stanley, did not analyse the evolved rules. Did the same sorts of rules evolve many times? Were there any general properties of these rules that emerged? Did they tend towards simple rules, or unique and/or complex rules for every evolutionary run and/or relatively fit individual in a run? How does modifying the complexity or other properties of the task affect the kinds of rules evolved?

3.1 Synaptic Plasticity Neuromodulation Models

The first model of neuromodulation of synaptic plasticity was introduced by Niv et al. [20]. The topology of the network was fixed and divided up into several modules. The plasticity of synapses between these modules and an output neuron could be gated by evolved dependencies on some of the other modules. If a module had a dependency on another module then synaptic plasticity was only enabled when neurons in the latter fired. They found that this gating of synaptic plasticity was required to evolve networks able to perform a simple reinforcement learning task. While this result is interesting its general applicability is not clear as the allowable dependencies were hand-crafted along with the network topology.

A more general model of neuromodulation of synaptic plasticity was introduced by Soltoggio et al. [26]. In this model there are two neuron types, standard and modulatory. Each neuron maintains values for both standard and modulatory activation, which are the weighted sums of all inputs from the two respective subsets of neurons in the network. Unlike a standard neuron model, the modulatory activation level modulates the plasticity of all synapses leading into the neuron by being used as an overall multiplicative term in the plasticity rule. Evolved solutions using this model outperformed, and could handle more complex environments than, solutions using the model developed by Niv et al. Additionally, solutions using models where neuromodulation was disabled or fixed weights were used either performed very poorly or could not be found at all. Later work showed similar results with a different task (double T-maze with reward moved at random points during trials) [25]: evolution produced solutions with much higher average performance using models incorporating neuromodulation of plasticity compared to models without this neuromodulation.

Similar to the model introduced by Soltoggio, Risi and Stanley [23] developed a model in which some connections, rather than neurons, are modulatory. In this model every neuron has a neuromodulatory activation level (as well as the standard activation) which modulates the plasticity of all synapses leading into the neuron. This model could be considered to more closely mimic biological networks as the same neuron can emit standard signals as well as modulatory signals.

Similar to synaptic plasticity models in general, an open question is what impact more flexible or complex neuromodulation properties would have on the evolution of plastic neural networks for online learning. For example it is known that in biological neural networks neuromodulators can invert the timing-dependency of STDP rules. Specifically, other avenues of potential inquiry include studying the use of multiple neuromodulators affecting different aspects of plasticity, less targeted neuromodulation models such as a neuromodulator that can be dispersed into a region of a network that is defined in some space, and the generation of modulation signals as a by-product of any neuron rather than designated modulatory neurons, similar to the operation of those in biological neural networks.

4 Synaptic Scaling/Competition Models and Neuronal Excitability Regulation Models

Di Paolo [8] studied the effect of explicit activity-dependent synaptic scaling (ADS) and "directional damping" applied to synaptic strength updates (DD). ADS induces homeostasis by actively scaling synaptic strengths in order to maintain post-synaptic firing rates within a pre-defined range. This is achieved by multiplying all weight values by the same factor, introducing heterosynaptic competition. DD adds a factor to the synaptic update rule such that if a change in strength of a synapse that is already near the limit would push the strength closer to the limit then the change is dampened, whereas a change that moves the strength away from the limit is not dampened. This type of damping was chosen for two reasons: it tends to produce a uni-modal distribution of weights centred around the point where potentiation and depression equilibrate, as opposed to other methods that either apply a simple hard limit or non-directional damping and which produce a bimodal distribution where most synapses either become fully potentiated or depressed; and because this model is supported by empirical evidence from studies of biological networks. The study compared the performance of evolved solutions incorporating ADS and/or DD. It was found that DD made no observable difference in the final performance achieved, but that ADS slightly increased it. It is not clear if the difference may have been reduced with longer evolutionary runs as only final performance rather than plots of performance over generations are indicated. Additionally, the reliability of the solutions using different models was assessed by introducing relatively rapid synaptic strength decay factors. Only solutions using ADS were able to perform reliably in this scenario; it is argued that this is a *"consequence of the compensatory nature of the ADS mechanism, which is able to alter synapses as a consequence of longer term changes in neural activity in ways that tend to maintain this activity."*

In later work Di Paolo [30] used another active neuronal excitability regulation mechanism to induce homeostatic firing rates. This mechanism enables synaptic plasticity when neurons are firing outside two pre-defined ranges and modifies the plasticity rule to push the firing rates back into the correct range by modifying the sign of plasticity rule parameters. It was found that non-homeostatic networks made more errors, and that the errors did not follow a predictable pattern, as compared to homeostatic networks which exhibited fewer errors, and for which the errors were more predictable.

Hoinville et al. [14] studied a method of heterosynaptic competition that keeps the squared sum of synaptic strengths equal to unity. Additionally, a symmetric odd activation function – which satisfies the centre-crossing condition and ensures that the operating range of each neuron is centred on the most sensitive region of its activation function [17] – was used to help induce neuronal excitability regulation. The aim was to achieve homeostasis without using active parameter manipulation, that is *"the neuronal activity is not monitored and there is no mechanism that dynamically corrects any parameter. In fact, homeostasis is not ensured to be maintained in the short- or long-term. However ... the chosen constraints would make homeostasis more likely ..."* Unlike the models introduced by Di Paolo, this approach avoids defining ranges for firing rates a priori. Either of these mechanisms in isolation was found to improve the final performance of solutions and the convergence rate during evolution, and especially so when used in tandem. Behavioural stability and robustness, measured by increasing the duration of the task to ten times longer than that used during evolution, was also found to be improved by use of the heterosynaptic competition mechanism and again especially so with both mechanisms. It is concluded that synaptic normalisation supports multistability by *"contributing to a global self-organization of individual plastic rules" and that "homeostasis can evolve implicitly without any active homeostatic mechanisms and be implemented through constrained hebbian plasticity."*

Di Paolo and Hoinville et al. have explored several synaptic scaling/competition and neuronal excitability regulation mechanisms in the context of online adaptation. Future research could study these mechanisms in the context of online learning, and explore other mechanisms found in biological neural networks, such as spike adaptation, where the firing threshold of a neuron is increased if high firing rates are maintained.

5 Encoding of Neural Networks in Evolved Genomes

Methods to encode neural networks into genomes for evolution can be broadly categorised into direct and indirect schemes. Direct encoding schemes employ a one-to-one mapping from elements in the genome to components in the phenotype, and include bit-string representations [3, 4, 10], vector-of-values [8, 9, 14, 20, 25, 26, 30], and graph-based schemes [21,27,29], where vertices in the graph correspond to neurons and edges to connections between them.

Indirect encoding schemes include a cellular encoding/grammar tree system [13], a matrix rewriting system [11], a "neural map" scheme where a vertex can either become a single neuron or a set of neurons and edges on the map vertex type may describe a one-to-one connection or one-to-all connections [29], HyperNEAT [22] and Evolvable-Substrate-HyperNEAT (ES-HyperNEAT) [23]. In HyperNEAT neurons exist in a geometric space, and a function, which is the genome, maps from the coordinates of a pair of neurons to the parameters of the synapse between them (including whether or not it exists). Thus to decode a network the coordinates of each and every pair of neurons is fed into the function and the outputs are used to determine the properties of the synapse between each pair. In this way the evolved genome function can encode the parameters of the network with respect to the geometry of the network, which can be advantageous when the input and/or output of the network contain geometrically encoded regularities [22, 23]. In ES-HyperNEAT the density and placement of connections, and so neurons, is determined by the amount of information encoded in each region of the hypercube represented by the genome function. Indirect encoding schemes can allow searching a smaller genome space while still creating complex networks [23].

Two developmental schemes have been used to evolve neural networks with plastic synapses. Gruau [13] developed a grammar tree system where the values of weights as determined by plasticity rules could be passed onto new connections created by recursive expansion of the grammar tree during an individuals life. Khan and Miller [15] devised a system where neurons, axons and dendrites situated in a Cartesian space could replicate, migrate, grow/shrink and terminate during an individuals life, partially dependent on environmental input. Each of these developmental functions was controlled by programs evolved using Cartesian genetic programming.

Evolution is the method of choice, and perhaps the only known method, for generating artificial neural networks based on models other than the traditional rate-based, fixed-weight variety (for which there are many methods, the most well known of which is likely back-propagation). There is a wide variety of encoding schemes described here, but relatively little information on the impact of using different encoding schemes on evolving neural networks for online learning.

6 Discussion and Future Directions

While relatively little research has been performed in total over the last couple of decades, advances have been made in tackling increasingly difficult online learning problem domains, by way of increasing the functional capabilities of neural network models and via improvements in methods of artificial evolution as a vehicle to generate neural networks for online learning. Of course, in the context of the goal of creating an agent capable of general intelligence, there is still a long way to go.

Problem domains studied to date have been relatively simple, as compared to many studied in the field of reinforcement learning, and also with respect to the goal of creating agents with some form of general intelligence. The environments typically represent a small state-, percept- and action space, with a small change occurring in the environment during an agents life time (for example, the location of a reward occasionally switches between two places in a fixed maze layout, or stimulus/reward associations are modified). Significantly, this small change is the only kind of change that occurs throughout an entire evolutionary run, and typically an entire set of experiments. These kinds of simple dynamic tasks require evolving a specialised exploration strategy and a specialised memory system to remember which fixed behaviour pattern is the current correct one or to remember a few simple associations. Additionally, all of the studied problem domains using a reinforcement learning paradigm have very clear exploration and exploitation phases: it is clear when exploration can cease and exploitation begin, either initially or upon the environment having changed; thus the specialised exploration strategy is invoked only when the expected reward for the current behaviour pattern or stimulus is not met. These have probably been desirable features, given available computational resources or a focus on other aspects of evolving plastic networks. However, an important question is whether plastic networks can be evolved to operate in more complex environments and, perhaps more importantly, to operate in environments that are significantly different to those seen at any other time during an evolutionary run. In other words, is it possible to evolve plastic neural networks that implement more general exploration strategies and can form and exploit internal models of more complex environments which have not previously been encountered?

As far as the authors are aware there have been no comparative studies on encoding schemes for plastic networks. An open question is what impact encoding schemes have on the ability to evolve plastic networks for the task of online learning.

Several studies have found that introducing functional properties observed in biological neural networks into the neural network models employed for online learning or adaptation, even in a very simplified form, has produced higher quality solutions, enabled new capabilities, and/or improved evolvability in terms of speed and reducing variance in performance. Examples include neuromodulation of synaptic plasticity [25, 26], synaptic scaling/competition [8, 14] and spike-timing-dependent plasticity [8].

Given these findings, and the many functional properties of biological neurons and synapses that have been implicated in learning, memory and information processing, an important question is what functional properties can be introduced to positive effect, particularly in the domain of online learning, for which only a relatively simple form of plasticity neuromodulation has been studied to date. Examples of functional properties for which there is evidence of a role in performing various kinds or aspects of online learning in biological neural networks include spike-timing-dependent synaptic plasticity [2, 5], synaptic scaling and competition [1], stochastic synaptic transmission [24], meta-plasticity [2], and synapto- and neuro-genesis [7].

On a similar note, both the neural map and the HyperNEAT-based schemes aim to allow generating regular structures in the network that map to certain kinds of regularities over multiple inputs and/or outputs [22, 23, 29]. Biological networks achieve this goal by self-organisation processes [7, 12] which can continue to operate throughout the networks life. Thus another interesting open question is whether emulations of these life-time plasticity processes in evolved artificial neural networks can facilitate online learning for such input or output spaces.

A factor of the success in tackling increasingly difficult online learning problem domains is the availability of computational power. Specifically, the availability of increasing computational power has made it possible to simulate more complex neural network models, perform longer evolutionary runs, and simulate more complex environments that require online learning. This factor should not be underestimated: evolution of artificial neural network models can be particularly computationally intensive, especially so for the task of online learning, which by its nature tends to require long fitness evaluation times in the evolutionary algorithm and the simulation of neural network models that are more complex than typical models. We are entering a period where this approach is becoming increasingly practical.

References

1. Abbott, L.F., Nelson, S.B., et al.: Synaptic plasticity: taming the beast. Nature Neuroscience 3, 1178–1183 (2000)
2. Abraham, W.C.: Metaplasticity: tuning synapses and networks for plasticity. Nature Reviews Neuroscience 9(5), 387–399 (2008)
3. Baxter, J.: The evolution of learning algorithms for artificial neural networks. Complex Systems, 313–326 (1993)
4. Chalmers, D.J.: The evolution of learning: An experiment in genetic connectionism. In: Proceedings of the 1990 Connectionist Models Summer School, pp. 81–90 (1990)
5. Cooke, S.F., Bliss, T.V.P.: Plasticity in the human central nervous system. Brain 129(7), 1659–1673 (2006)
6. Cuccu, G., Gomez, F.: When Novelty Is Not Enough. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcázar, A.I., Merelo, J.J., Neri, F., Preuss, M., Richter, H., Togelius, J., Yannakakis, G.N. (eds.) EvoApplications 2011, Part I. LNCS, vol. 6624, pp. 234–243. Springer, Heidelberg (2011)
7. Deng, W., Aimone, J.B., Gage, F.H.: New neurons and new memories: how does adult hippocampal neurogenesis affect learning and memory? Nature Reviews Neuroscience 11(5), 339–350 (2010)
8. Di Paolo, E.: Spike-timing dependent plasticity for evolved robots. Adaptive Behavior 10(3-4), 243–263 (2002)
9. Di Paolo, E.: Evolving spike-timing-dependent plasticity for single-trial learning in robots. Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 361(1811), 2299–2319 (2003)
10. Floreano, D., Mondada, F.: Evolution of plastic neurocontrollers for situated agents. In: Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., Wilson, S. (eds.) From Animals to Animats, vol. 4. MIT Press, MA (1996)
11. Floreano, D., Urzelai, J.: Neural morphogenesis, synaptic plasticity, and evolution. Theory in Biosciences 120(3), 225–240 (2001)

12. Fuerst, P.G., Burgess, R.W.: Adhesion molecules in establishing retinal circuitry. Current Opinion in Neurobiology 19(4), 389–394 (2009)
13. Gruau, F., Whitley, D.: Adding learning to the cellular development of neural networks: Evolution and the baldwin effect. Evolutionary Computation 1(3), 213–233 (1993)
14. Hoinville, T., Siles, C.T., Hénaff, P.: Flexible and multistable pattern generation by evolving constrained plastic neurocontrollers. Adaptive Behavior 19(3), 187–207 (2011)
15. Khan, G.M., Miller, J.F., Halliday, D.M.: Evolution of cartesian genetic programs for development of learning neural architecture. Evolutionary Computation 19(3), 469–523 (2011)
16. Kistemaker, S., Whiteson, S.: Critical factors in the performance of novelty search. In: Proceedings of the 13th annual Conference on Genetic and Evolutionary Computation (GECCO 2011), pp. 965–972 (2011)
17. Mathayomchan, B., Beer, R.D.: Center-crossing recurrent neural networks for the evolution of rhythmic behavior. Neural Computation 14(9), 2043–2051 (2002)
18. McHale, G., Husbands, P.: Quadrupedal locomotion: GasNets, CTRNNs and hybrid CTRNN/PNNs compared. In: Proceedings of the 9th International Conference on the Simulation and Synthesis of Living Systems (Alife IX), pp. 106–112 (2004)
19. Mouret, J.B., Tonelli, P.: Artificial evolution of plastic neural networks: a few key concepts. In: DevLeaNN (2011)
20. Niv, Y., Joel, D., Meilijson, I., Ruppin, E.: Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. Adaptive Behavior 10(1), 5–24 (2002)
21. Risi, S., Hughes, C.E., Stanley, K.O.: Evolving plastic neural networks with novelty search. Adaptive Behavior 18(6), 470–491 (2010)
22. Risi, S., Stanley, K.O.: Indirectly Encoding Neural Plasticity as a Pattern of Local Rules. In: Doncieux, S., Girard, B., Guillot, A., Hallam, J., Meyer, J.-A., Mouret, J.-B. (eds.) SAB 2010. LNCS, vol. 6226, pp. 533–543. Springer, Heidelberg (2010)
23. Risi, S., Stanley, K.O.: A unified approach to evolving plasticity and neural geometry. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2012). IEEE (2012)
24. Seung, S.: Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. Neuron 40(6), 1063–1073 (2003)
25. Soltoggio, A., Bullinaria, J.A., Mattiussi, C., Dürr, P., Floreano, D.: Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. Artificial Life 11, 569 (2008)
26. Soltoggio, A., Dürr, P., Mattiussi, C., Floreano, D.: Evolving neuromodulatory topologies for reinforcement learning-like problems. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 2471–2478 (2007)
27. Stanley, K.O.: Evolving adaptive neural networks with and without adaptive synapses. In: Proceedings of the 2003 Congress on Evolutionary Computation, vol. 4, pp. 2557–2564 (2003)
28. Tonelli, P., Mouret, J.: Using a map-based encoding to evolve plastic neural networks. In: 2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS), pp. 9–16. IEEE (April 2011)
29. Tonelli, P., Mouret, J.-B.: On the relationships between synaptic plasticity and generative systems. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011, pp. 1531–1538. ACM, New York (2011)
30. Wood, R., Di Paolo, E.A.: New Models for Old Questions: Evolutionary Robotics and the ‘A Not B’ Error. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 1141–1150. Springer, Heidelberg (2007)