Towards autonomous developmental artificial intelligence: case study for explainable AI.

STARKEY, A. and EZENKWU, C.P.

2023

Towards Autonomous Developmental Artificial Intelligence: Case study for Explainable AI

Andrew Starkey[1[0000-0003-1797-8737]] and Chinedu Pascal Ezenkwu[2[0000-0001-7232-8441]]

[1] Aberdeen University, UK. a.starkey@abdn.ac.uk
[2] Robert Gordon University, UK. p.ezenkwu@rgu.ac.uk

Abstract. State-of-the-art autonomous AI algorithms such as reinforcement learning and deep learning techniques suffer from high computational complexity, poor explainability ability, and a limited capacity for incremental adaptive learning. In response to these challenges, this paper highlights the TMGWR-based algorithm, developed by the present authors, as a case study towards self-adaptive unsupervised learning in autonomous developmental AI, and makes the following contributions: it presents and reviews essential requirements for today's autonomous AI and includes analysis for their potential for Green AI; it demonstrates that, unlike these state-of-the-art algorithms, TMGWR possesses explainability potentials that can be further developed and exploited for autonomous learning applications. In addition to shaping researchers' choice of metrics for selecting autonomous learning strategies, this paper will help to motivate further innovative research in autonomous AI.

Keywords: Autonomous AI, Green AI, unsupervised learning.

1 Introduction

A great deal of effort has been invested in autonomous artificial intelligence utilising high-performing machine learning algorithms, such as deep learning. However, these techniques continue to suffer from high computational costs, lack of explainability, and a limited capacity for incremental adaptive learning – which can lead to catastrophic forgetting and lack of self-recoverability in dynamic contexts. With the increasing adoption of AI in the global economy, a responsible application of Green AI techniques (with reduced carbon footprint) has become an important consideration alongside performance. In addition, there have been reports of catastrophic failures leading to casualties in the use of self-driving cars (Schmelzer, 2021). Therefore, to achieve a safe autonomous intelligent system, the system should be computationally efficient and meet Green AI criteria, be adaptable to changes in its environment, and should provide explainable outputs.

This study is amongst AI papers recommending a strategy based on the unsupervised learning paradigm as the way towards truly autonomous AI (LeCun, Bengio and Hinton, 2015; Marcus, 2018). The Temporospatial Merge Grow When Required (TMGWR) network has recently been proposed to address the challenges of self-organising approaches and can compete favourably with traditional reinforcement learning in autonomous agent behaviours when incorporated with value iteration (Ezenkwu and

Starkey, 2019, 2022). TMGWR and other self-adaptive unsupervised mechanisms are suitable for autonomous agents and neurorobotics due to their ability to support lifelong learning (Tenzer, Rasheed and Shafique, 2022).

While previous research into TMGWR has demonstrated that it is sample-efficient, self-adaptive, and can cope with unpredictable scenarios through dynamic planning, its explainability potential has not been explored further. With our belief in self-adaptive unsupervised learning as an effective approach to true autonomous agents, this paper makes the following contributions:

1. it reviews essential requirements for today's autonomous AI.
2. it demonstrates that, unlike reinforcement learning, TMGWR can be developed to give explainable outputs to a human observer that can be further developed and exploited for important autonomous learning applications.

Although this paper emphasises TMGWR, it is due to its superiority over other sensorimotor map learning strategies as presented in our previous paper (Ezenkwu and Starkey, 2019). Moreover, TMGWR is only a case study for reviewing the potential of self-adaptive unsupervised learning as well as discussing general desiderata for today's autonomous AI.

2 Requirements for today's autonomous Artificial Intelligence

Despite the significant successes recorded by sophisticated AI algorithms such as deep learning and reinforcement learning, they are neither safe nor suitable for autonomous learning due to the following reasons: (a) the environmental risks associated with high-performing but expensive AI techniques, (b) the inflexibility of these techniques due to their data-hungry nature, (c) their lack of explainability due to their black box architectures. These challenges can be presented under categories such as data intensiveness, task inflexibility, explainability, bias, and societal integration.

Based on the above challenges with the state-of-the-art AI techniques, the goal of all autonomous AI research should be to realise an AI technique with the potential to address these limitations as detailed in the following sections.

2.1 Computational efficiency of learning algorithms

Some sophisticated AI techniques have revealed promising results in different areas of application. For example, in 2015 DeepMind Technologies developed AlphaGo, a deep reinforcement learning algorithm that became the first to defeat a master in the game of Go (Silver et al., 2016); deep learning has also proven to be a popular method in different classes of AI problems such as computer vision, natural language processing (NLP), self-driving cars and so on (LeCun et al., 2015). Another example is that despite being reported as the best performer in natural language processing (NLP) tasks in terms of accuracy (Edwards, 2021; Wang, Niu, Zhao, Wang, Hao and Che, 2021), the carbon dioxide emissions of training and applying a transformer are even more substantial than the lifetime emissions of an automobile (Strubell, Ganesh and McCallum, 2019).

The success of AlexNet in the 2010 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has demonstrated that the depth of a deep learning model is significant for its high performance (Krizhevsky, Sutskever and Hinton, 2012), hence leading to competition in achieving the deepest neural networks in the field. The number of parameters a deep neural network has appears to correlate with the performance accuracy, with deeper neural network architectures emerging in recent times. For example, with 175 billion parameters and 96 total layers, GPT-3 has been ranked the best in NLP (Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, Askell et al., 2020). Similarly, EfficientNet-L2 has been ranked the best performing image classifier to date, with 480 million parameters trained on 130 million images (Xie, Luong, Hovy and Le, 2020).

These requirements raise clear concerns around the cost and the carbon footprint due to the use of energy-intensive hardware (Strubell et al., 2019; Anthony, Kanding and Selvan, 2020; Justus, Brennan, Bonner and McGough, 2018) and do not meet requirements for a Green AI.

Table 1. Summary of computational costs of learning algorithms. N=number of observations, D=vector size, M=number of hidden neurons, O=number of output values, C=number of clusters, S=size of state space, P=population size, G=size of chromosome

Learning algorithm	Time complexity	Order of complexity
SVM with Newton	N^3	$O(n^3)$
Feed-forward neural networks	D.M.O.N	$O(n^4)$
Decision tree	N^2.D	$O(n^3)$
K-means	C.N.D	$O(n^3)$
SOM	N.D	$O(n^2)$
Growing SOM	N.D	$O(n^2)$
Q-learning	S^3	$O(n^3)$
Value iteration	S^2	$O(n^2)$
Genetic algorithms	P.log(P).G	$O(n.\log(n^2))$

Table 1 summarises the time complexities of the most common learning algorithms (Kearns, 1990; Koenig and Simmons, 1993; Nicolas, 2017; Kearns, Vazirani and Vazirani, 1994). The computational complexities of algorithms are crucial in deciding which best fits a given scenario.

SOM and growing SOM have computational advantage over most of the methods in Table 2. The number of nodes affects the speed of the algorithm, with the correct number of nodes crucial for improving the efficiency and representational ability of the algorithm. For the two planning algorithms in Table 1, model-based RL algorithms such as value iteration have a better time complexity than model-free RL algorithms such as Q-learning. Unlike Q-learning, which does not know the effect of an action before it is executed at least once, value iteration only needs to enter a state at least once to discover all of its successor-states (Koenig and Simmons, 1993). However, the model of an environment may not always be available or constant for that environment — in these cases, model-based RL will fail.

2.2 Self-adaptation

A desirable attribute of an autonomous system is the ability to cope with changing scenarios, especially when the environment is unpredictable. Because the real world is very complex and uncertain, it is probable that an agent designer will not capture all the possibilities of a given task during design time.

For example, a notorious consequence of lack of self-adaptation is the catastrophic failure of self-driving cars (Schmelzer, 2021). Self-driving cars make judgments based on their perceptions of their surroundings and pre-set traffic rules (Kang, Zhao, Qi and Banerjee, 2018). As a result, road construction, traffic signal failures, challenging weather conditions, confusing parking signs, and further unimagined circumstances could cause a self-driving car to fail (Kang et al., 2018). Gheibi et al have studied the extent to which different machine learning paradigms have been applied to self-adaption tasks (Gheibi et al, 2021), and shown that reinforcement learning (RL), specifically model-free RL, is the most used learning method for self-adaption, followed by supervised learning, with little attention paid to unsupervised learning.

However, each of these popular methods has inherent problems that can restrict their self-adaptivity. For example, RL agents require a task-dependent reward function. A reward function is an indirect way in which the agent designers infuse their domain expertise into the design. The design of the reward function often requires understanding of the environment and can pose a challenge if the agent's world changes in a manner not anticipated during the design. Unlike RL agents, supervised learning agents require explicit provisions of ground truths by a teacher, meaning that they need humans in the loop to adapt to new changes. So, with a slight change in a task, a RL or supervised learning agent may need modification in its learning mechanism and retraining to overwrite the previous knowledge for the new task. This problem is known as catastrophic forgetting (Kirkpatrick et al., 2017).

2.3 Explainability of learning algorithms

One perspective adopted in the explainability of a learning algorithm is transparency (Belle and Papantonis, 2021). Transparency is the extent to which a human can understand the learning mechanisms of an algorithm (Lipton, 2018). It is possible to achieve the explainability of an already trained model through post-hoc processing (Tan et al., 2020; Belle and Papantonis, 2021). Since post-hoc processing adds computational overheads to the process of training and deploying a learning algorithm, this paper favours an inherently transparent learning model.

Belle and Papantonis present and compare three aspects of learning transparency — simulatability, decomposability, and algorithmic transparency (Belle and Papantonis, 2021). Simulatability is a model's ability to be replicable in human thought.

Decomposability is the ability of a human to break down a model into inputs, parameters and computations and then explain these parts, while algorithmic transparency is the ability of a human to understand and explain the mechanism by which a model generates its output. While complex methods such as SVM, ensemble learning and multilayer neural networks are opaque and require post-hoc processing for explainability, simpler methods such as logistic regression, K-nearest neighbour algorithm, and rule-based learners are inherently transparent and do not require any post-hoc processing to understand the model.

In addition to the models considered in the paper by Belle and Papantonis, SOM and other neighbourhood-based algorithms such as K-means algorithm and Vector Quantisation have been considered transparent (Tan et al., 2020; Aliyu, 2018). While the transparency of evolutionary algorithms such as genetic algorithms is dependent on the cost function they are meant to optimise, they have been used for post-hoc processing (Pickering and Cohen, 2021). RL has the same explainability issue as evolutionary algorithms. Explainable RL, especially in complex environments, is an open research question (Kuhnle, May, Schafer and Lanza, 2021). Works by Belle and Papntonis (Belle and Papantonis, 2021) and Chazette et al (Chazette, Brunotte and Speith, 2021) have provided in-depth reviews on explainability.

2.4 Summary of desired learning attributes

The above sections describe the problems with current approaches; they are computationally intensive and do not satisfy Green AI, cannot adapt to changes, and cannot explain their decision making processes. These attributes then form the goals for a truly autonomous method: low compute satisfying Green AI; adapting automatically to changes; and explainable to the human operator.

3 Case study of self-adaptive unsupervised learning: TMGWR

A sensorimotor map is an agent's self-model of the world. Using the learned sensorimotor map of an environment or an agent's experiences, the agent can exhibit autonomous behaviours — either self-motivated or goal-directed. Previous work described the limitations of unsupervised learning approaches for sensorimotor map learning, such as Connectionist World Model (CWM) (Toussaint, 2004) using Growing Neural Gas (GNG) (Fritzke, 1995). This motivated the proposal of the TMGWR (Ezenkwu and Starkey, 2019), which is an adaptive neural architecture that learns the topological map and the sensorimotor links (Butz et al., 2008) between neurons using a time series self-organising strategy (Strickert and Hammer, 2005a). The TMGWR network connects nodes based on their sensorimotor proximities, such that these edges can encode the transition possibilities as well as the motor signals that can cause transitions between nodes.

The experiment for this case study was designed on a simple maze environment, with a randomly changing goal within this maze and also new walls being added to the maze

on a random basis. Although this is a simple problem, current solutions require human design and highlight the lack of adaptive learning properties described above.
The performance of the TMGWR-based agent with those of model-free and model-based RL agents was compared (Ezenkwu and Starkey, 2022). Although the TMGWR-based agent can be classified as a model-based RL agent, it differs from the traditional model-based agent in the sense that instead of requiring a human designer to encode the dynamics of the environment, it self-learns its world model using the TMGWR algorithm. The work by the authors evaluated the algorithms' sample complexities and their abilities to self-adapt to a sudden change in the environment or goal state. This work has shown that the TMGWR network gave more efficient representations of the environment in a computationally more efficient manner, and that it also showed the potential for online adaptation to changes in goal state or changes in the environment. However, the approach did not meet the requirements for Explainability as discussed in earlier sections.

The TMGWR framework consists of four main modules - the sensorimotor map learning module, the sensory preprocessor, the motivation estimator and the action selector (Ezenkwu and Starkey, 2022). An autonomous agent is equipped with suitable sensors and actuators which enable it to observe the environment and react to these observations using the actuators. The observations or sensory inputs can be preprocessed or transformed into a form that conveys meaningful or contextual information to the agent. The preprocessed sensory observations are passed on to the sensorimotor map learning module which enables the agent to develop or refine its mental model of the scenario. The sensorimotor map learning occurs continually in an open-ended manner to enable the agent to keep track of changes in the environment by continuously updating the sensorimotor map. The motivation estimator provides the motivation signal that enables the agent to plan towards a goal or behave in a given manner in the environment. The action selector considers the current observation and the agent's motivation in selecting the best action using the sensorimotor map. Execution of this action causes a change in the environment and the cycle continues.

3.1 Sensorimotor map learning

This section examines the TMGWR algorithm as a sensorimotor map learning method. The key features of TMGWR are that:

- the nodes are linked based on their sensorimotor proximities to one another;
- it uses the temporal context vector similar to Merge Grow Neural Gas (MNG) (Strickert and Hammer, 2005b) to keep track of the sensorimotor history;
- the GWR strategy of adding new nodes is employed to enable the system to keep track of changes in the environment;
- all the hyperparameters are kept constant throughout the lifetime of the agent to encourage continual learning.

The action map learns the codebook vector for each motor activity while the sensorimotor map learns the input weight vectors and the possible action vectors linking them to each other. At each time step, the activated action vector on the action map is associated with the sensorimotor-link from the previous winning node *i* to the current winning node *j* in the sensorimotor map. The algorithm uses a similarity function that compares the activated action vector at a given time with the action vector that has already been associated with the sensorimotor-link from node *i* to node *j*. This similarity function has been chosen to be a Gaussian function so that if the two activation vectors are similar then it tends towards 1 otherwise it will tend towards 0.

The advantage of introducing this similarity function in the update equation is that it increases the weight of the sensorimotor-link if the same action vector results in the same transition all the time and decreases it if the transition is possible with different action vectors. This modification is motivated by Hebbian associative learning, which reinforces the association between two neurons that fire together and discourages those which do not (Frolov and Murav'ev, 1993). This is a way of representing reliability in the agent's mental model and it is useful during planning as the agent is more likely to select reliable actions for each experience in the environment.

A full description of the TMGWR algorithm is presented in in Ezenkwu and Starkey, 2022 and the reader is directed there for further information on the algorithm.

3.2 Suitability for autonomous learning

Compared to both model-free and model-based RL agents, the TMGWR-based goal-directed agent has proven to be far more self-adaptive in situations of changing environment or changing goal state. In addition, the experiments demonstrated that the TMGWR-based algorithm shows a similar sample complexity to the model-based RL agent but is better than the model-free RL agent. The TMGWR-based agent requires less time to self-adapt to changing goal states than the model-free RL agent and a change in the environment, than the other algorithms, with the model-based agent being completely intolerant to a slight change in the environment.

Short demonstrations of the change of goal scenario for the Model-free RL[1], TMGWR-based[2] and the Model-based RL[3] agents, and for the responses of the Model-free RL[4], TMGWR-based[5] and model-based RL[6] agents to dynamic environments are available on YouTube for view.

---

1 Demonstration: response of the model-free RL agent to change in goal state: https://www.youtube.com/watch?v=_j0z6B1RFjs

2 Demonstration: response of the TMGWR-based agent to change in goal state: https://www.youtube.com/watch?v=x9U0r-6Sct0

3 Demonstration: response of model-based RL agent to change in goal state: https://youtu.be/4GNbxYvJPhM

4 Demonstration: response of the model-free RL agent to change in the environment: https://youtu.be/aRr4Ja9TspQ

5 Demonstration: response of the TMGWR-based agent to change in the environment: https://youtu.be/-YpxGEjRoXA

6 Demonstration: response of model-based RL agent to change in the environment: https://www.youtube.com/watch?v=peEYriVEK2k

3.3 Explainability in the TMGWR-based algorithm

This section discusses the development of the explanation mechanism of the TMGWR-based algorithm, exploiting the transparent nature of the algorithm and thereby how it can provide explanations to a human observer.

The TMGWR-based agent makes decisions using the sensorimotor map. An effective sensorimotor map should be able to represent the reality of the agent's environment. One benefit of this representation is that the agent can anticipate the outcome of its actions. For example, if the sensorimotor map represents that the agent takes action, a_{ik}, at node i, its next state will be a node, k, then the agent can anticipate this next state each time action, a_{ik}, is to be executed at node i. If the environment does not change as expected following this action, then this means that the sensorimotor map no longer represents the reality of the agent's world and can imply a change in the environment. This then represents the main contribution of this paper and the changes made to the algorithm that permits feedback to the human observer in terms of any change in the environment or goal state, since any expected change in the environment that is not met means that the TMGWR's internal representation of the environment is no longer current and will require to be changed. This change can be communicated to the human observer and more importantly can be described in terms of the actions and environment states that the algorithm expected to take place. As an example, if a new wall is introduced into the maze, then the previous learning will predict that the agent can move into the space now occupied by the wall. The algorithm will detect the lack of change in the sensor values (i.e. position in the maze) following the action having been taken (i.e. move to space now occupied by the wall). This can immediately be communicated to the human observer by the agent: I expected to be able to move forward; the world has changed since I cannot.

Therefore, the procedure for keeping track of a change in the environment is as follows:

- after selecting action a_{ik} at the current node i;
- use the sensorimotor map to anticipate the next state node k as follows: $k = argmax_n V(n)$, for all node n in the sensorimotor neighbourhood of node i, while $V(n)$ is the motivation potential at node n;
- execute action a_{ik} and identify the actual resulting node r;
- if k does not equal r (i.e. a different node has been activated) then the world has changed;
- otherwise, no observable change in the world.

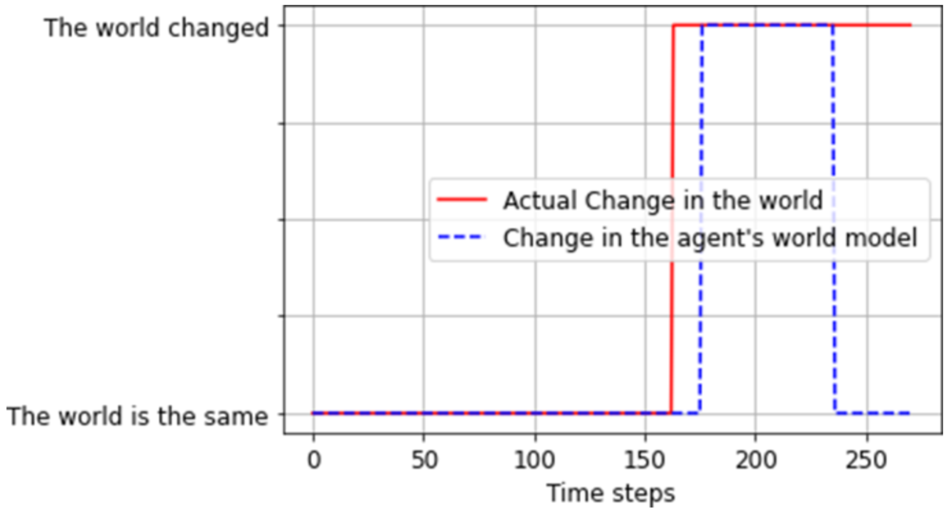

Fig. 1. TMGWR-based agent's response to a change in the world.

Fig 1 demonstrates how a TMGWR-based agent responds to a change in the environment. After the agent's world changes, the agent will lose its ability to anticipate the outcomes of its actions. To the agent, these anticipation failures mean that the world has changed. However, the agent will self-adapt its mental model to correctly anticipate the outcomes of its actions to cope with the new state of the world. The rate of change for the TMGWR algorithm depends on a single parameter that can be modified so that it is more sensitive to changes in the environment, otherwise it will gradually change its model so that eventually it will choose a different action in order to reach its goal. The TMGWR approach thereby meets the requirements for explainability in terms of......

Learning paradigms such as RL and supervised learning lack this explainability potential because their world models are not interpretable. A video demonstration of this procedure has been provided here[7].

Explanation mechanism in the TMGWR-based agent during a change of goal.

The TMGWR-based algorithm uses a motivation estimator to compute motivation potentials of all the nodes in the sensorimotor map. The motivational potential of a node is a function of how similar the node is to the goal node and the availability of sensorimotor links from that node to the goal node. Therefore, after each run of the motivation estimator, the goal node will always have the highest motivation potential because it is the most similar to itself and the most easily reachable from itself. Based on this, the TMGWR-based agent can keep track of a change in goal by computing motivation potentials after each step and identifying any changes in the node with the largest motivation potential.

[7] Demonstration: Change in environment: https://www.youtube.com/watch?v=CSooq2abq4g

Fig 2 demonstrates how the sensorimotor map due to TMGWR can therefore immediately and correctly reflect a change in goal following the above procedure. There is no straightforward way of realising a similar interpretation of goal change in RL agents. The video demonstration can be found in this link[8].

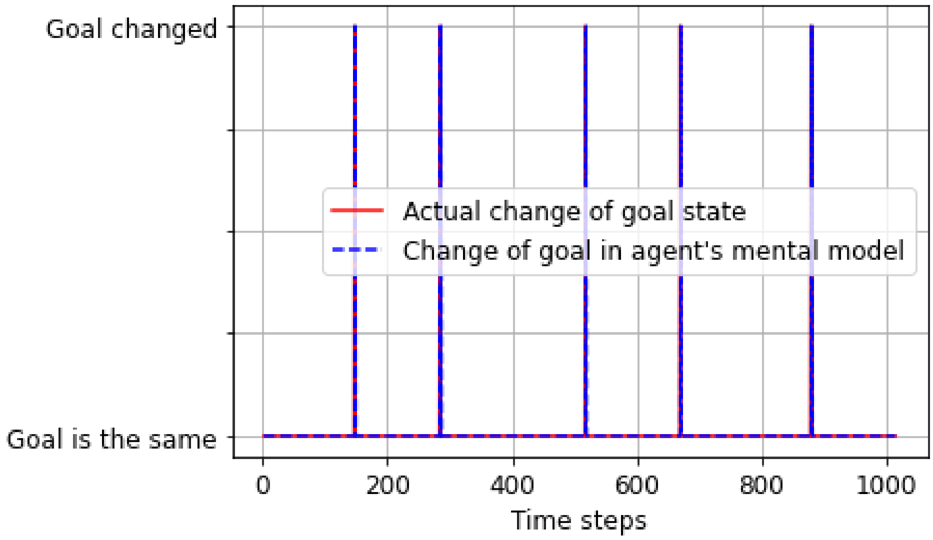

Fig. 2. Identification of change in goal state using TMGWR-based approach

4 Conclusions

This paper has highlighted the requirement for a learning framework that is self-adaptive, sample efficient, and requires less compute power thereby meeting requirements for Green AI. Currently popular AI techniques such as deep learning and deep reinforcement learning are sample inefficient, inflexible, and require significant designer input and huge compute power and so do not meet these requirements. The paper reviews different learning algorithms with respect to considerations such as computational efficiency, self-adaptivity and explainability. Based on this review, growing SOM has been identified as the most suitable learning paradigm for future autonomous learning agents. It is unsupervised, self-adaptive and inherently transparent and has an efficient computational cost when compared to popular methods such as deep learning, SVM and ensemble learning. The paper recommends an autonomous agent architecture based on the TMGWR network for continuous sensorimotor map learning and shows how improvements to this meet the requirements above by giving an effective demonstration of the explanation potential of the TMGWR-based framework for changes in goal and also changes in environment. Future work will focus on applying the

[8] Demonstration: Change in goal - https://www.youtube.com/watch?v=Mv0s79CFBtI

TMGWR approach to more sophisticated environments and to different domain problems.

References

1. Aliyu, A.U., 2018. Automated data classification using feature weighted self-organising map (fwsom). University of Aberdeen, PhD Thesis.
2. Anthony, L.F.W., Kanding, B., Selvan, R., 2020. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. arXiv preprint arXiv:2007.03051 .
3. Belle, V., Papantonis, I., 2021. Principles and practice of explainable machine learning. Frontiers in big Data , 39.
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Nee- lakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165 .
5. Butz, M.V., Reif, K., Herbort, O., 2008. Bridging the gap: Learning sensorimotor-linked population codes for planning and motor control, in: International Conference on Cognitive Systems, CogSys.
6. Chazette, L., Brunotte, W., Speith, T., 2021. Exploring explainability: A definition, a model, and a knowledge catalogue, in: 2021 IEEE 29th International Requirements Engineering Conference (RE), IEEE. pp. 197–208.
7. Edwards, C., 2021. The best of nlp. Communications of the ACM 64, 9–11.
8. Ezenkwu, C.P., Starkey, A., 2019. Unsupervised temporospatial neural architecture for sensorimotor map learning. IEEE Transactions on Cognitive and Developmental Systems 13(1), 223-230
9. Ezenkwu, C.P., Starkey, A., 2022. An unsupervised autonomous learning framework for goal-directed behaviours in dynamic contexts. Advances in Computational Intelligence 2, 1–14.
10. Fritzke, B., 1995. A growing neural gas network learns topologies, in: Advances in neural information processing systems, pp. 625–632.
11. Frolov, A., Murav'ev, I., 1993. Informational characteristics of neural networks capable of associative learning based on hebbian plasticity. Network: Computation in Neural Systems 4, 495–536.
12. Gheibi, O., Weyns, D., Quin, F., 2021. Applying machine learning in self-adaptive systems: A systematic literature review. arXiv preprint arXiv:2103.04112 .
13. Justus, D., Brennan, J., Bonner, S., McGough, A.S., 2018. Predicting the computational cost of deep learning models, in: 2018 IEEE international conference on big data (Big Data), IEEE. pp. 3873–3882.
14. Kang, L., Zhao, W., Qi, B., Banerjee, S., 2018. Augmenting self-driving with remote control: Challenges and directions, in: Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications, pp. 19–24.
15. Kearns, M.J., 1990. The computational complexity of machine learning. MIT press.
16. Kearns, M.J., Vazirani, U.V., Vazirani, U., 1994. An introduction to computational learning theory. MIT press.
17. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al., 2017. Over-coming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences 114, 3521–3526.

18. Koenig, S., Simmons, R.G., 1993. Complexity analysis of real-time reinforcement learning, in: AAAI, pp. 99–107.
19. Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25, 1097–1105.
20. Kuhnle, A., May, M.C., Schafer, L., Lanza, G., 2021. Explainable reinforcement learning in production control of job shop manufacturing system. International Journal of Production Research .
21. LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436–444.
22. Lipton, Z.C., 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue 16, 31–57.
23. Marcus, G.,2018. Deep learning: A critical appraisal. arXiv preprint arXiv:1801.00631.
24. Nicolas, P.R., 2017. Scala for Machine Learning: Data processing, ML algorithms, smart analytics, and more. Packt Publishing Ltd.
25. Pickering, L., Cohen, K., 2021. Toward explainable ai—genetic fuzzy systems—a use case, in: North American Fuzzy Information Processing Society Annual Confer- ence, Springer. pp. 343–354.
26. Schmelzer, R., 2021. What happens when self-driving cars kill people? URL: https://www.forbes.com/sites/cognitiveworld/2019/09/26/ what-happens-with-self-driving-cars-kill-people/.
27. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., 2016. Mastering the game of go with deep neural networks and tree search. nature 529, 484–489.
28. Strickert, M., Hammer, B., 2005a. Merge som for temporal data. Neurocomputing 64, 39–71.
29. Strickert, M., Hammer, B., 2005b. Merge som for temporal data. Neurocomputing 64, 39–71.
30. Strubell, E., Ganesh, A., McCallum, A., 2019. Energy and policy considerations for deep learning in nlp. arXiv preprint arXiv:1906.02243 .
31. Tan, R., Khan, N.M., Guan, L., 2020. Locality guided neural networks for explainable artificial intelligence. CoRR abs/2007.06131. URL: https://arxiv.org/abs/ 2007.06131, arXiv:2007.06131.
32. Tenzer, M., Rasheed, Z., Shafique, K., 2022. Learning citywide patterns of life from trajectory monitoring. arXiv preprint arXiv:2206.15352 .
33. Toussaint, M., 2004. Learning a world model and planning with a self-organizing, dynamic neural system, in: Advances in neural information processing systems, pp. 926–936.
34. Wang, L., Niu, D., Zhao, X., Wang, X., Hao, M., Che, H., 2021. A comparative analysis of novel deep learning and ensemble learning models to predict the allergenicity of food proteins. Foods 10, 809.
35. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V., 2020. Self-training with noisy student improves imagenet classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10687–10698.