An Analysis of Current Continual Learning Algorithms in an Image Classification Context

1st Jess Wakelin
Centre for Smart and Advanced Technologies (CAST)
University of Northampton
Northampton, UK
jess.wakelin19@my.northampton.ac.uk

2nd Noor Abdalkarem Mohammedali
Centre for Smart and Advanced Technologies (CAST)
University of Northampton
Northampton, UK
noor.mohammedali@northampton.ac.uk

***Abstract*—Artificial Intelligence aims to mimic natural intelligent learning by using lifelong-machine-learning, which allows an AI to train and learn over it's lifetime. Various algorithms have been suggested and developed to allow lifelong learning, these algorithms require deeper analysis, to evaluate and highlight performance benefits. In this research, we will study with three state-of-the-art algorithms for lifelong learning: Rehearsal, elastic-weight-consolidation and synaptic-intelligence. We do an analysis and evaluation of their performance in a multiple task experiment, using different amounts of data, measuring several performance metrics. We found that these algorithms are similar in performance, but some algorithms perform better than others with less data, or show good performance in task one, but not subsequent tasks. These algorithms could be built upon and improved in future research. The evaluation demonstrated in this research are in the image classification context.**

***Index Terms*—Continuous Learning, Image Classification, Lifelong Machine Learning , Non-Continuous Learning, Machine Learning, Artificial Intelligence, Intelligent Agent**

I. Introduction

Artificial Intelligence (AI) is commonly used across many facets of daily life and numerous technology will use AI to make decisions without human input. These decisions will be made based on the previously collected data and this can generally be formed into two categories of prediction: classification and regression [1]. Classification prediction tasks are most common and assign the agent a data instance. The agent's experience is used to predict where the data-point falls into predefined categories, for instance, predicting cat, breed from a photograph [2].

Regression prediction tasks the agent with optimising complex systems involving linear and nonlinear multivariate regression problems [2], for instance, to predict the price of a given property over the coming decade.

Thus far, the typical method for machine learning is aptly named: train-use-replace. This method of machine learning defines the life-cycle of an agent as follows: the agent is designed and trained using a curated dataset; the agent is put to work, solving the problems for which it was trained; finally the agent is replaced with a newly developed model, trained with a newly developed dataset [3], [4]. This life-cycle means that a model cannot change and grow during its working lifetime, there is no opportunity for repairs or improvements.

Recent research demonstrates that it is possible to introduce continuous updates and training to the AI during operation, bringing benefits that include extension of lifespan and improvement in performance, and several algorithms have been developed to accomplish this. Continuous learning could revolutionise AI technology improving effectiveness and flexibility across all applications.

Plenty of artificial intelligence techniques, including image classification, utilise deep-learning [1], [5]. Deep-learning is defined as the process by which a model independently forms the rules that place a particular data-point into its class as this is not defined by the developers [3]. This process is used for image classification as manually defining these patterns can be exceptionally difficult.

Lifelong-machine-learning (LML) is defined by [5] as a sequence of N tasks used to train AI to do a particular set of tasks during the lifetime. Also, further training is allowed as explained by [1] to get multiple outcomes. Several algorithms have been develop to suit various LML applications and the main objectives are listed below [5], [6]: 1- To improve performance by doing more training on current tasks. 2- To expand the domain of the model by introducing new classes or new tasks.

New Instances (NI) scenario is where new data is added to the model which fits into familiar classes, this data is structured in a similar way to the initial training set to support the existing model with further training. This is especially suitable for rare cases found so that the model expanded to perform more reliably [6].

New Instances and Classes (NIC) is adding new data to the model which does not fit into familiar classes, this means that the domain of the models task is expanded, which can improve utility [6].

New Tasks scenario is where the model is trained to perform a task that is unfamiliar and, ideally, isolated from its original task, meaning that the task domains do not overlap. With careful training a model can perform well in many tasks [6].

The aim of this research is to demonstrate some algorithms devised for NI and New Task scenarios to assess the differences between algorithms using deep-learning methods. Performance, cost and training dataset requirements will be considered. This analysis will illustrate the existing situation of LML and the other areas will be highlighted in this field to do more studies on it to improve its performance.

In Section II, the related work on development of lifelong machine learning is summarized, and used to select an appropriate dataset and the LML algorithms. In Section III, we experiment to evaluate and analyse the selected algorithms. In Section IV, we discuss comparable solutions and evaluate performance. Finally, we conclude and highlight future work in Section V.

II. Literature Review

Typically, a classification model is trained on a single task and will perform only this task for its lifetime [7]; this means that the training process, and training data, need to be completed in full before deployment and that model cannot be modified during deployment, replacement is required.

This constitutes the train-use-replace life-cycle, devised to maximise performance before the advent of LML. Achieving high reliability in an AI has been shown to have many difficulties; safety-critical applications necessitate scrutiny of an AI, as a false prediction could be dangerous. Adoption of the train-use-replace life-cycle has enabled developers to guarantee that the agent sustains acceptable performance throughout its lifetime, introduction of LML creates risk of the agent becoming less competent, thereby increasing the chances of failure, unacceptable in safety-critical scenarios. Safety-critical applications can include: Medical diagnostics [4] and Autonomous vehicles [8], the risk of an agent's performance declining discourages use of LML in these fields.

Some AI, including image classification models [3], utilise deep-learning [1], [5]. Deep-learning allows a model to independently form the rules for patterns that place a particular data-point into it's class as this is not defined by the designer [3].

Deep-learning generally accelerates model architecture periods and encourages better model performance for such data as images or video-feeds [3]. This is achieved by a removal of designer participation in the class definition process, allowing the model to perform this at lower cost and higher reliability. One such model type designed for deep-learning is a Convolutional Neural Network (CNN), which will be discussed in this research.

LML is an approach to model training that can allow the model to continuously learn, independently, to perform tasks outside of its initial domain [1], [5]. LML can therefore be utilised to improve model performance, utility or lifespan.

The LML research aims to develop a method of training that allows an agent to emulate a natural intelligence [1], in that it can learn new information or skills without limit and improve performance in these skills through 'practice'. Potential applications are explained in [6] as one of these types: new tasks, new instances (NI) and new instances and classes (NIC).

LML comes with two major problems beyond what is typically seen by an AI, both being related to the longer length of training that can be undertaken by the LML algorithm. These problems are called overfitting and catastrophic forgetting, both characterised by poor prediction accuracy. Overfitting is the process wherein a model, in training for a task, becomes too fixated on finer details in instances of the training data [3]. This gives the model a tendency to focus on details and debilitates the models ability to recognise more general patterns, causing the model perform badly on unseen data [3]. This is typically characterised as good training performance and significantly poorer testing performance [3].

A model is typically constituted by a series of weights that define how it makes predictions. Each weight may be more important to particular predictions than others [5], [9] and training a model means brute-force and trial-and-error modification of weights to achieve better performance [3].

Catastrophic forgetting (CF) is the phenomenon where a model's performance may degrade with training, this can be caused by changing weights in training for a new task where these weights are important to a previous task, and so the new task knowledge effectively overwrites old knowledge, causing performance of old tasks to degrade [1], [4]–[6], [9].

Methods to address and alleviate CF can reduce changing of the weights that are most relevant to an old task, by preventing change, limiting change or reversing change of the weights involved in the performance degradation [5], [9], [10], or some combination of these three. In performing this it is critical to know which weights are important to old tasks, so that these weights can be selected and controlled [5], [9], [10].

An analysis of a competition conducted by [11], where competitors demonstrated LML solutions utilising computer vision, suggests that current LML algorithms are sufficient for deployment in video-stream applications. The analysis, however, also found that over-engineered solutions where commonplace and manual performance evaluations of solutions are taxing.

Future research into Evolved Plastic Artificial Neural Networks (EPANN) are encouraged by advances in neural network research and developments in computational power. EPANN is a machine learning method designed to emulate natural intelligence, with the aim to autonomously create new learning schemes as detailed by [12].

III. Proposed Experiment

To improve our understanding of existing continual learning algorithms, we propose an experiment to objectively measure comparable metrics for each algorithm. Data collected in this experiment will allow for direct comparison between algorithms. The experiment should consist of a test that is applicable to all the algorithms to enable fair comparison of results.

Algorithms selected for this experiment are: Rehearsal, elastic-weight-consolidation (EWC) and synaptic-intelligence (SI). The reasons for this decision are discussed in this section. Rehearsal, as it's name suggests, is designed to retrain for old tasks during training for new tasks, to reverse the effects of catastrophic forgetting. EWC and SI both aim to regularise weight changes, to reduce catastrophic forgetting, during training. These three algorithms compared to a Naive approach, and this algorithm is not designed for continual learning; this will highlight the utility of specialised algorithms.

Google Colab services will be used to execute the experiment. Outsourced GPU training should reduce the required time span for the experiment, this outsourcing is made possible by Colabs service, which provides remote access to a virtual machine with allocated GPU.

The proposed experiment necessitates a considerable volume of data, this will introduce a long training time, so the dataset should minimise the size of each data instance to mitigate this effect. The selected dataset is MNIST [13], a dataset composed of 60,000 28x28 greyscale images of handwritten Arabic numerals, this dataset is open-source and suited to AI applications. Data for new tasks will be a created by permuting MNIST. MNIST was chosen as it has a considerable volume of small data instances. A large number is necessary to enable the dataset to be split into differently sized sections with a significant margin, so that the the performance difference visible between dataset size used for training is beyond margin-of-error.

Small size of data instances is a crucial feature of the dataset to mitigate the increased training time caused by a larger volume of instances. Due to the small size of each instance, the data within its capacity will be more primitive, thereby making it not necessarily representative of real-world applications, however, this should not matter; The tested machine-learning algorithms act as a modification of a given CNN, meaning that the absolute results are less important than the difference in results between algorithms. This difference in results should be replicable when the tested LML solutions are applied to another scenario, for example, a more complex CNN and dataset, representative of real-world applications.

Training methods for each algorithm were specially designed for this experiment; data collection during training is required to enable evaluation and analysis, training processes have been designed to effectively facilitate this collection. Data collected for each test is: for each size of data set, for each task, for each epoch, training-loss, loss, accuracy, total training time in seconds and final accuracy.

Each training process is modeled as a series of nested loops to ensure that training can be performed in various configurations can be tested and measured, the results are saved in Dictionary datatype.

Results are graphed using the collected data, each graph will include one line for each LML algorithm to enable accurate comparison and will show various axes configurations so that all the data can contribute to analysis.

[14] demonstrates examples of Naive and Rehearsal algorithms and these have been largely appropriated for this experiment; we hope to be able to understand in more detail some of the intricacies and challenges of LML by testing a wider array of algorithms.

A Naive approach to LML problems has been shown by [5] to be largely ineffective; Naive is used in this experiment as a 'control sample' to demonstrate the necessity of specialised algorithms. Naive refers to a 'Traditional' machine learning approach when applied to scenarios better suited to specialised LML algorithms, including NIC or new task scenarios. Naive, being by definition not suited to LML, is likely to demonstrate poor performance in this experiment.

[14] also demonstrate EWC, this implementation was appropriated and modified for this experiment. The modifications covered the training process to better suit the nature of the experiment, which necessitates testing on various sizes of dataset. Synaptic-intelligence (SI) is designed as a direct modification come upgrade to EWC [5], which is thought to be the most effective algorithm for this application, and as such will serve as a good comparison.

SI, like EWC, is designed to regularise weights when training, specifically, after the initial batch [5]. Regularisation reduces CF by limiting weight changes when these weights are important to previous tasks. SI differs from EWC in its calculations for regularisation.

The SI was implemented in Colab for this experiment, as an existing solution was not found to be available. SI and EWC are similar in construction and as such, the implementation of SI was formed to keep coherence with the EWC implementation [14].

$$L = L_{\text{cross}}\ \left(\widehat{\mathbf{y}}, \mathbf{t} = \widehat{\mathbf{y}}_{1\ \text{h}}\right) + \frac{\lambda}{2} \cdot \sum_{k} F_k \left(\theta_k - {\theta_k}^*\right)^2 \tag{1}$$

$$\Delta L_k = \Delta\theta_k \cdot \frac{\partial L}{\partial \theta_k} \tag{2}$$

$$F_k = \frac{\sum \Delta L_k}{T_k^2 + \xi} \tag{3}$$

Zenke et al. proposed the EWC implementation in [10], and then in 2019, Maltoni et al. worked on this implementation [5]. Now, in this research, we are going to use the same algorithms for experimentation.

The mechanisms of SI are described in these equations.

As SI is a modification of EWC [5], start with the equation to calculate the EWC Loss. The loss function of the EWC described in equation 1. L_{cross} is cross-entropy loss and $\hat{y}$ is network predictions, which evolve as the model changes, but it is out of our scope for now.

λ is the regularisation constraint (constant). θ_k denotes the value of a given weight and θ_k* describes the optimal weight value from the previous task.

F_k is the *k*th element of the fisher matrix, which describes the importance of the weight θ_k. [10] theorised that calculating the fisher matrix is expensive and unnecessary and developed SI by calculating F_k directly as opposed to reading from the matrix, hence giving performance gains.

This was achieved using equation 3. $\Sigma\Delta L_k$ is defined as the total change in loss over update steps of one weight as is shown in equation 3 and calculated using equation 2.

T_k^2 is described as the total movement of weight θ_k during training on one batch and ξ is a small constant, 1×10^{-3} for [10], to avoid division by zero.

This means that F_k, as described in the equation 3, can be substituted into the Loss calculation as is shown in equation 1, eliminating the process of calculating the complete fisher matrix, forming the substance of the loss algorithm which characterises Synaptic Intelligence (SI).

IV. Result and Discussion

A. SIT Scenario

The SI compared and demonstrated to a naive New Instances (NI) in a single incremental task (SIT) application as is shown in Fig 1. This comparison shows that SI is not suited to an SIT scenario.

The two methods display similar results across all metrics, though SI shows consistently slightly better performance, although this variation is mostly within the margin of error.

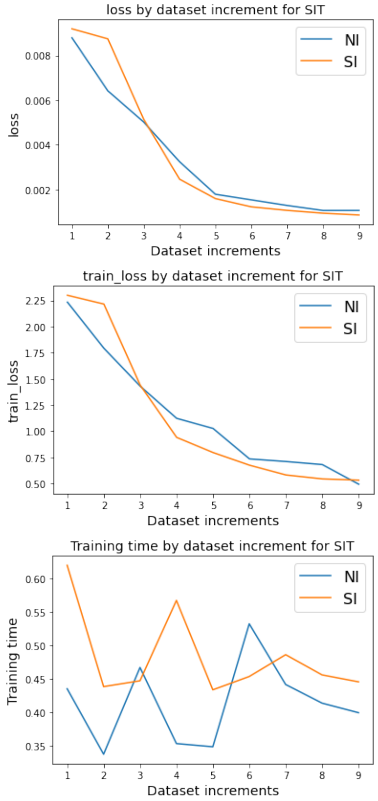

Fig. 1. Graphed Results for SIT

In terms of performance, a remarkable difference is visible in smaller dataset increments, showing NI performing better at this end of the spectrum. Both algorithms display weak performance in small dataset increments, as there was little training data used, although this could be mitigated with more epochs by the training stage.

Furthermore, SI regularly suffers in training time compared to NI. With the single incremental task schema, non-continuous learning CNN is better for new instances rather than continuous learning through SI.

B. Continuous Learning Algorithms - Metrics by Epoch

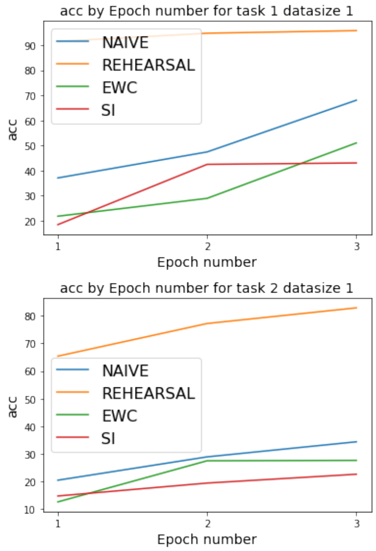

Fig. 2. Accuracy by epoch for datasize 1

Accuracy, shown in Fig 2 is generally weaker for later tasks, task two being the weakest across algorithms with three algorithms hovering around twenty percent, poor performance when a random guess would produce ten percent accuracy. Rehearsal is however shown to be an outlier, superior to other algorithms, consistently in excess of sixty-five percent.

Despite exhibiting best performance, rehearsal suffers from a drop in accuracy in later tasks, similar to other algorithms, reaching a maximum value of ninety-five percent in task one, but achieving only up to eighty-five percent across two and three.

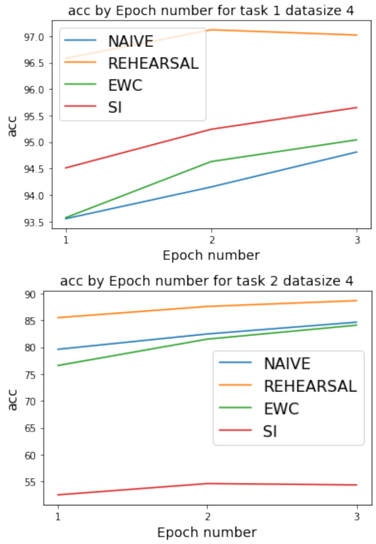

Fig. 3. Accuracy by epoch for datasize 4

Moving to dataset size 4, performance has improved across the board, as expected. The differences between algorithms encourage variance in performance, which emerges at this dataset size. Rehearsal continues to be a the best performing, as shown in Fig 3, increasing task one performance by two percent and tasks two and three performance by five percent. All algorithms however have approached the performance of rehearsal in all tasks, with exception of SI, who is amongst other models in task one, but far behind in other tasks in terms of accuracy.

Thus far, in this earlier stage, rehearsal shows a high propensity for learning, but begins to plateau in learning rate around stage four, with task one performance reaching this point slightly earlier than other tasks, allowing competitors to approach.

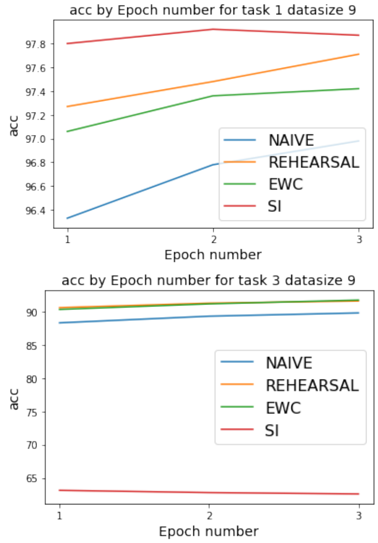

Fig. 4. Accuracy by epoch for datasize 9

This trend continues until the final stage of the experiment: dataset size nine, where, as before, there is a great improvement in performance in each algorithm, rehearsal has reached a plateau and other algorithms are approaching, except for SI which, curiously, performs best in task one, boasting approximately ninety-eight percent accuracy, slightly better than rehearsal, but by far weakest in tasks two and three, achieving under sixty-five percent in task three, as evidenced in Fig 4.

C. Continuous Learning Algorithms - Metrics by Task

Fig 5 illustrates, like previous examples Fig 2 and 3, rehearsal has a high learning gradient in earlier stages, reaching high accuracy in earlier dataset sizes averaging eighty percent in size one, and improving gradually for the remainder of training. Rehearsals competitors average between twenty and forty percent accuracy in dataset size one, notably below rehearsal, but begin to close the gap moving into dataset size four, with the exception of SI task two and three, which are, as before, significantly weaker in performance.

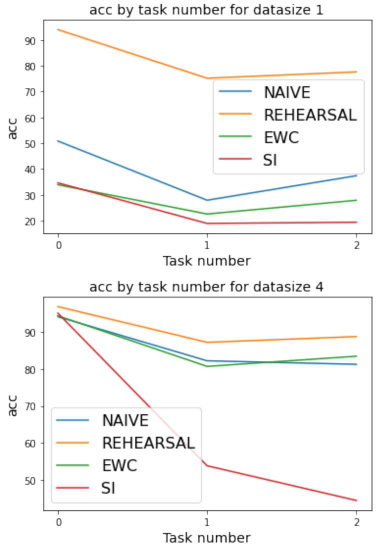

Fig. 5. Accuracy by task number

Across all three sizes of the used dataset, we notice that EWC and Naive get a great benefit when the size of the dataset increases. The performance of EWC and Naive between dataset sizes one and four jumps more than our expectation and this jump is not displayed by SI and rehearsal. On the other hand, from dataset sizes four to nine the performance shows minor improvement across all tasks, whereas rehearsal approaches that plateau even earlier.

SI demonstrated improvement in it's performance, especially in tasks two and three; This suggests that SI may require a bigger dataset to approach a plateau, similar to other algorithms, beyond dataset size nine.

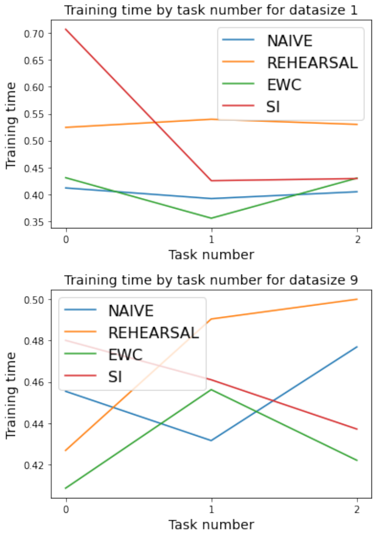

Fig. 6. Training time by task number

The training time is shown in Fig 6 which is measured for each model along three epochs. The patterns here are significantly differed from other metrics.

The time measured shows a lack of variance between algorithms and between dataset size used, although one pattern that is notable is a consistently higher training time for rehearsal. This is as expected, repetitive training is used by the rehearsal algorithm which means that extra training is needed per task, meaning more time is taken. SI initially shows a significantly increased time measured, possibly an anomalous result.

Achieving success in this research means reaching a verdict on the details of, or lack of, performance benefits, introduced by LML algorithms, when applied to an applicable scenario. Ideally, the tested approaches will show at least one beneficial aspect in an experimental setting, implying that LML could prove useful under applicable circumstances. Beneficial aspects can include, but are not limited to: better final accuracy, better accuracy using only certain volumes of training data, rate of performance gain in seconds, minimum training data to achieve a predefined performance threshold, vulnerability to known machine learning problems including overfitting and catastrophic forgetting.

V. Conclusion and Future Work

The experiment has demonstrated the advantages and the power of each LML approach. Rehearsal algorithm appears to have the best performance across all tasks competing with other algorithms when using a small dataset. SI performance was extraordinary in the first task then the performance decreased in subsequent tasks. When the dataset size increased, the performance increased as well which gives a good indication that it can be better than a Naive CNN. For that reason, as previously discussed, the Naive CNN is not suitable for the LML scenarios. With new advancements, the LML will prove useful for many applications, for example, medical imaging, where the use of the LML would aid in maximising reliability by teaching the model of new medical research as it arises. The LML will also prove useful for autonomous vehicle applications when dealing with new obstacle types, to maintain the performance of the algorithm.

This paper highlights the LML field and the significant progress made by the researchers in the literature and how we can use it in computer vision, for example, video streaming or autonomous vehicles. Future research will work on untested scenarios to show the capability of the LML with various algorithms using similar techniques to do the experiment. Furthermore, computational power needs to be considered when we work with the LML to accelerate the speed of learning in an advanced way.

References

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[2] M. G. Alalm and M. Nasr, "Artificial intelligence, regression model, and cost estimation for removal of chlorothalonil pesticide by activated carbon prepared from casuarina charcoal," *Sustainable Environment Research*, vol. 28, no. 3, pp. 101–110, 2018.

[3] H. H. Aghdam and E. J. Heravi, "Guide to convolutional neural networks," *New York, NY: Springer*, vol. 10, no. 978-973, p. 51, 2017.

[4] M. Perkonigg, J. Hofmanninger, C. J. Herold, J. A. Brink, O. Pianykh, H. Prosch, and G. Langs, "Dynamic memory to alleviate catastrophic forgetting in continual learning with medical imaging," *Nature Communications*, vol. 12, no. 1, pp. 1–12, 2021.

[5] D. Maltoni and V. Lomonaco, "Continuous learning in single-incremental-task scenarios," *Neural Networks*, vol. 116, pp. 56–73, 2019.

[6] V. Lomonaco and D. Maltoni, "Core50: a new dataset and benchmark for continuous object recognition," in *Conference on Robot Learning*, pp. 17–26, PMLR, 2017.

[7] N. Gupta and R. Mangla, *Artificial Intelligence Basics: A Self-Teaching Introduction*. Mercury Learning Information, 2020.

[8] B. Wilson, J. Hoffman, and J. Morgenstern, "Predictive inequity in object detection," *arXiv preprint arXiv:1902.11097*, 2019.

[9] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[10] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*, pp. 3987–3995, PMLR, 2017.

[11] V. Lomonaco, L. Pellegrini, P. Rodriguez, M. Caccia, Q. She, Y. Chen, Q. Jodelet, R. Wang, Z. Mai, D. Vazquez, *et al.*, "Cvpr 2020 continual learning in computer vision competition: Approaches, results, current challenges and future directions," *Artificial Intelligence*, vol. 303, p. 103635, 2022.

[12] A. Soltoggio, K. O. Stanley, and S. Risi, "Born to learn: the inspiration, progress, and future of evolved plastic artificial neural networks," *Neural Networks*, vol. 108, pp. 48–67, 2018.

[13] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[14] A. Carta, "Intro to continual learning," May 2021.