Continual Learning with Deep Streaming Regularized Discriminant Analysis

Joe Khawand [1,2] Peter Hanappe [2] David Colliaux [2]

[1] Ecole Polytechnique
[2] Sony Computer Science Laboratories Paris
joe.khawand.20@polytechnique.org

Abstract

Continual learning is increasingly sought after in real-world machine learning applications, as it enables learning in a more human-like manner. Conventional machine learning approaches fail to achieve this, as incrementally updating the model with non-identically distributed data leads to catastrophic forgetting, where existing representations are overwritten. Although traditional continual learning methods have mostly focused on batch learning, which involves learning from large collections of labeled data sequentially, this approach is not well-suited for real-world applications where we would like new data to be integrated directly. This necessitates a paradigm shift towards streaming learning. In this paper, we propose[1] *a streaming version of regularized discriminant analysis as a solution to this challenge. We combine our algorithm with a convolutional neural network and demonstrate that it outperforms both batch learning and existing streaming learning algorithms on the ImageNet ILSVRC-2012 dataset.*

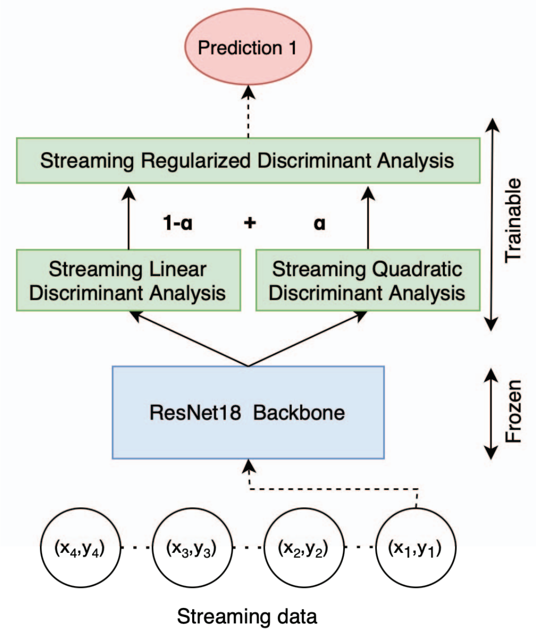
Prediction 1
Streaming Regularized Discriminant Analysis
1-α
+
α
Streaming Linear Discriminant Analysis
Streaming Quadratic Discriminant Analysis
Trainable
ResNet18 Backbone
Frozen
(x4,y4)
(x3,y3)
(x2,y2)
(x1,y1)
Streaming data

Figure 1. Deep SRDA model diagram.

1. Introduction

Continual learning, also known as lifelong learning, refers to the ability of a learning system to sequentially acquire and adapt knowledge over time. This type of learning mimics animal learning [8] and is increasingly sought after in various domains such as medical diagnostics [22], autonomous vehicles [40], and finance [35], where the learner needs to continually adapt to changing data. The major challenge in continual learning is the phenomenon of *catastrophic forgetting* [9, 31]. It refers to the situation where a naively incrementally trained deep neural network forgets previously learned representations to specialise to the new task at hand.

Traditionally, the bulk of research [45] in continual learning has primarily concentrated on batch learning approaches, which process data in fixed batches. In this setting, a continual learner typically iterates multiple times over the given task in an offline manner, allowing them to achieve satisfactory performance. However, this approach requires storing all data from the current task for training, which is not suitable for on-device learning.

As a result, recent research has emerged in the field of Online Continual learning [27], where data arrives in small, incremental batches and previously seen batches from the current or previous tasks are no longer accessible. Therefore, a model must effectively learn from a single pass over the online data stream, even when encountering new classes (Online Class Incremental, OCI) or data non-stationarity, such as new background, blur, noise, illumination, and occlusion (Online Domain Incremental, ODI).

[1] https://github.com/SonyCSLParis/Deep_SRDA.git

We take this scenario one step further and consider a **streaming case of the online continual learning** scenario, where a learner learns from batches of size 1. This particular case of Streaming learning aims to develop methodologies that can efficiently learn from streaming data, enabling continuous adaptation without relying on complete batches. Specifically, we concentrate on the application of classification in computer vision, focusing on the general scenario of Online Class Incremental.

In this paper, we aim to contribute to the field of continual learning by proposing a novel approach called Deep Streaming Regularized Discriminant Analysis (SRDA). Building upon the foundations of Deep Streaming Linear Discriminant Analysis (SLDA) [14], our method combines SLDA with a streaming version of Quadratic Discriminant Analysis (QDA) to achieve state-of-the-art performance. As done in [14], we combine our model with a convolutional neural network (CNN) and empirically demonstrate its superiority over the other streaming and batch learning methods on the ImageNet ILSVRC-2012 dataset [39]. To the best of our knowledge, this use of a regularized discriminant analysis method with a neural network represents a novel contribution that has not been explored before.

This paper makes the following contributions:

1. We present the SQDA and SRDA algorithms. We show that SQDA does not generalize to high dimensional problems and present SRDA as a solution.

2. We demonstrate that SRDA outperforms state-of-the-art streaming and continual learning algorithms.

2. Related work

2.1. Continual Learning

Continual learning addresses the challenges of catastrophic forgetting that occur when training a model incrementally, breaking the usual i.i.d. assumption on the training data. This problem arises from the plasticity dilemma [1] and has been heavily studied in recent years [45]. Various continual learning scenarios have been developed to continually train models, with the three main ones being task incremental, class incremental, and domain incremental.

In the task incremental setting, different training steps are identified by a label. Models trained in this setting tend to perform well because there is an indication of the task in the data. However, this scenario is not very realistic as real-world data is not typically labelled by tasks. The class incremental scenario considers the real-world scenario of adding new classes without any task delimiters. Lastly, in the domain incremental scenario, the focus is on dealing with the addition of new domains or changing environments without any explicit task labels.

To mitigate catastrophic forgetting, several methods have been employed. The main ones include regularization [26, 44, 20, 49, 3, 23], rehearsal or pseudo-rehearsal [38, 41, 37], combined [18, 5, 24], and architectural [29, 25]. Notably, the rehearsal and pseudo-rehearsal categories have shown the most promising results. In these approaches, the learner stores previously encountered samples in a buffer for future training [38]. In some instances, pseudo-rehearsal techniques explore the replacement of the buffer with a generative model [41, 37].

2.2. Online Continual Learning

Online continual learning is a more challenging subset of continual learning where data arrives in an online fashion one tiny batch at a time and previously encountered batches are not accessible. This field builds upon existing methods for continual learning while adding specific tricks to tackle this scenario. In this online setting, recent works [30, 16, 47, 2, 27] have shown that the Softmax layer and its associated Fully-Connected layer suffer from *task-recency bias*, where those layers tend to be biased to the last encountered classes. This has prompted the creation of multiple tricks to alleviate this problem. One example is the application of various tricks in replay-based scenarios:

- **Labels Trick** [50]: Cross-entropy loss calculation considers only the classes present in the mini-batch, preventing excessive penalization of logits for classes absent from the mini-batch.

- **Multiple Iterations** [4]: A single mini-batch is stored in a buffer and iterated upon multiple times. In addition to that, previously stored experiments are also replayed.

- **Nearest Class Mean Classifier**: Replaces the last biased fully connected classification layer by a nearest mean classifier such as in iCarl [36].

- **Separated Softmax** [2]: Since one softmax layer results in a bias explained in [30, 16, 47, 2, 27], this technique employs two Softmax layers one for old classes and one for new classes. Thus training new classes will not overly penalize the old logits.

- **Review trick** [6]: Adds an additional fine-tuning step using a balanced subset of the memory buffer. This trick is used in the End-to-End method used in our benchmarks 5.3.

However, when the batch size is reduced to one, the Stochastic Gradient Descent (SGD) usually employed in most of these methods becomes noisy, making convergence challenging. This is precisely where streaming learning comes into play.

2.3. Streaming Learning

Streaming learning, a field of study since 1980 [33], primarily focuses on i.i.d data streams and utilizes online learning methods. However, due to the Softmax bias and SGD instability for batches of size 1, regular online learning methods are not optimal for streaming learning on **non-i.i.d data streams**.

For this case, one area of streaming learning considers the use of streaming decision trees [21], where Hoeffding decision trees [17] are adapted to avoid catastrophic forgetting. Those can also be combined into Streaming forests [21, 46]. However, the issue with these types of methods is that they are slow to train [11], and require extensive hyperparameter tuning, making them unsuited for fast-paced streaming scenarios and real-time on-device learning.

Another approach involves employing a Nearest Mean Classifier [32] instead of a Softmax layer. Research conducted by [28] has demonstrated that this simple yet effective substitute not only addresses recency bias but also avoids structural changes in the Fully-Connected layer when new classes are encountered. Notably, this method has been effectively employed by iCarl [36].

Another method used is Exstream [13]. This method only updates fully connected layers of a CNN while maintaining a prototype for each class. It also has a policy for managing the buffer when it is full, merging the two closest exemplars. But as we will see in section 5.5, this method suffers in terms of computation time as it requires, in this case, 64 hours to run on our experiment, whereas SRDA requires 12 hours.

Especially relevant to this paper is Streaming LDA [34] that was first used for data streams and has since been adapted in [14] to work with CNNs. SLDA uses running class means and a common covariance matrix for all classes to assign labels to inputs based on the closest Gaussian distribution.

3. Problem Setting

We consider ensembles $\mathcal{X}$ and $\mathcal{Y}$, representing our datapoints and labels, respectively. We aim to train a model F with parameters θ to accurately classify classes in $[\![1, C]\!]$, where $C \in \mathbb{N}^*$. To achieve this, we adopt a streaming fashion approach, where each datapoint $x \in \mathcal{X}$ is individually sent to the model for fitting. Additionally, we adopt a class incremental scenario by ordering the samples in batches of classes. We consider this type of scenario to be the most general as it is similar to animal and human learning scenarios.

4. Deep Streaming RDA

Similar to Hayes and Kanan's work [14], our model can be formally divided into a composition of two distinct functions G and F, such as $y = F(G(x))$. G is comprised of the initial layers of a CNN, specifically a ResNet-18 [15] in our case, while F represents our SRDA head. The early layers of a CNN, such as those in G, tend to learn filters that exhibit minimal variation across large natural image datasets and demonstrate high transferability [48]. Therefore, we made the decision to freeze the parameters of G and solely focus on training F.

The following subsections will present our SRDA model. We will start by presenting discriminant analysis before presenting an initial quadratic streaming version that led to our deep SRDA algorithm.

4.1. Discriminant Analysis

Discriminant analysis is a traditional machine learning algorithm that can be used for classification [12]. It works on the hypothesis that the data follows a Gaussian multivariate distribution that is used to calculate the log posterior probability using Bayes' rule.

For each training example $x \in \mathcal{X}$ and $k \in [\![1, C]\!]$, the goal is to calculate the posterior probability in order to classify correctly. The Bayes' rule on the posterior probability of being in class k for an element x is:

$$P(y = k|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y = k)P(y = k)}{P(\boldsymbol{x})} \tag{1}$$

With $P(\boldsymbol{x}|y = k)$ modeled as a multivariate Gaussian distribution with a mean μ_k and a covariance Σ_k:

$$P(\boldsymbol{x}|y = k) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^t \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)\right)}{(2\pi)^{C/2}|\boldsymbol{\Sigma}|^{1/2}} \tag{2}$$

According to equations 1 and 2, the log of the posterior or **discriminant** γ_k is given as follows:

$$\begin{aligned} \gamma_k = -\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^t \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k) \\ + \log P(y = k) + B \end{aligned} \tag{3}$$

Where $B \in \mathbb{R}$ is a constant.
Finally, the classification rule is written as:

$$F(\boldsymbol{x}) = \arg\max_k \gamma_k \tag{4}$$

With no further assumptions, this is referred to as Quadratic Discriminant Analysis (QDA). Linear Discriminant Analysis (LDA) and the streaming version of it [14], constrains equation 3 and considers equal covariance matrices between classes.

4.2. Streaming Discriminant Analysis

4.2.1 Quadratic

In order to adapt equation 3 to streams of data we need to calculate μ_k, Σ_k, and Σ_k^{-1} in a streaming fashion. We choose to replace those values by their **empirical estimators**.

We consider a new element z_t and $k \in [\![1, C]\!]$. The update functions are written as follows, where:

- c is the vector of encountered classes:

$$c_{(k=y,t+1)} = c_{(k=y,t)} + 1 \tag{5}$$

- $\hat{\mu}$ is the saved class means:

$$\hat{\boldsymbol{\mu}}_{(k=y,t+1)} = \frac{c_{(k=y,t)}\hat{\boldsymbol{\mu}}_{(k=y,t)} + z_t}{c_{(k=y,t)} + 1} \tag{6}$$

- $\hat{\Sigma}$ is the vector containing all the class covariance matrices:

$$\hat{\boldsymbol{\Sigma}}_{(k,t+1)} = \frac{t\hat{\boldsymbol{\Sigma}}_{(k,t)} + \boldsymbol{\Delta}_t}{t+1} \tag{7}$$

$$\boldsymbol{\Delta}_t = \frac{t(z_t - \hat{\boldsymbol{\mu}}_{(k=y,t)})(z_t - \hat{\boldsymbol{\mu}}_{(k=y,t)})^T}{t+1} \tag{8}$$

- Λ is the inverse of Σ regularized with a shrinkage coefficient ϵ:

$$\boldsymbol{\Lambda}_{(k,t)} = [(1-\epsilon)\hat{\boldsymbol{\Sigma}}_{(k,t)} + \epsilon\boldsymbol{I}]^{-1} \tag{9}$$

- $P(y = k)$ is calculated by incrementally and uniformly updating it for seen classes at time t. For a balanced dataset, this factor can be considered constant but is important for unbalanced datasets serving as a corrective term.

$$P(y = k)_t = \frac{c_{(k=y,t)}}{\sum_{n=1}^{C} c_{(k=n,t)}} \tag{10}$$

Applying those updates to equation 3 leads to a streaming version of QDA mentioned by Hayes and Kanan [14]. But the problem with SQDA is that, in high dimensionality, the number of datapoints needed to correctly empirically estimate the covariance matrices of each class are high [10]. As we will show in section 5, this approach struggles to translate to our high dimensional problem and mostly works with **low dimensional** datasets or ones with numerous examples per class. **This prompted us to look for a regularized alternative that solves this issue.**

4.2.2 Regularized

Friedman [10] proposed a compromise between LDA and QDA, that shrinks the separate covariances of QDA toward a common covariance as in LDA. Using a coefficient $\alpha \in [0, 1]$, the regularization targets the class covariance matrices as follows:

$$\bar{\boldsymbol{\Sigma}}_{(k,t)} = \alpha\hat{\boldsymbol{\Sigma}}_{(k,t)} + (1-\alpha)\hat{\boldsymbol{\Sigma}}_{t'} \tag{11}$$

Where $\hat{\Sigma}_{(k,t)}$ is the empirical class covariance calculated through QDA (eq.7), and $\hat{\Sigma}_{t'}$ the empirical covariance matrix calculated with SLDA, in this equation 12:

$$\hat{\boldsymbol{\Sigma}}_{t'+1} = \frac{t'\hat{\boldsymbol{\Sigma}}_{t'} + \boldsymbol{\Delta}_{t'}}{t'+1} \tag{12}$$

Replacing this new regularised $\bar{\Sigma}$ in equations 9 and 3 gives us **SRDA.**

The coefficient α controls the degree of shrinkage of the individual class covariance matrix estimates towards the pooled estimate. Since it is often the case that even small amounts of regularization can largely eliminate quite drastic instability [43], some values of α have the potential of superior performance when the population class covariances substantially differ [10]. This performance boost is clearly shown in section 5.

5. Experiments & Results

5.1. Baselines

We conducted a comprehensive analysis by comparing our method with both streaming methods and batch streaming methods. To evaluate the performance of our model, we utilized the metric described in [13, 19], which involves normalizing a model's performance by the offline model's performance:

$$\Omega_{all} = \frac{1}{T}\sum_{t=1}^{T}\frac{\rho_t}{\rho_{offline,t}} \tag{13}$$

In our case, ρ_t refers to the top-5 accuracy of our model at time t.

An Ω_{all} of 1 indicates that the continual learner performs equally well as the offline model. While it is theoretically possible to achieve results higher than one if the continual learner outperforms the offline model, such instances are rare in practice.

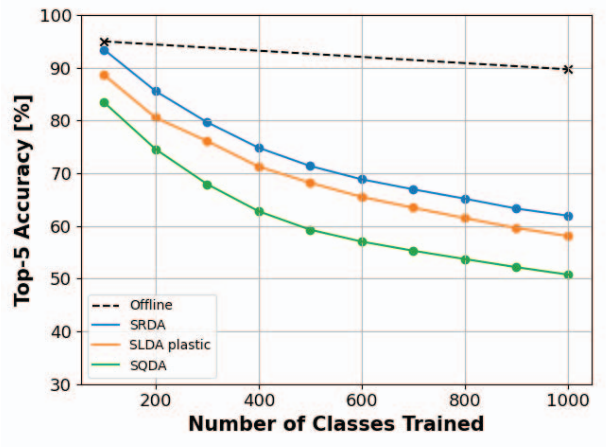

Figure 2. Top-5 Accuracy on ImageNet ILSVRC-2012. We compare our SRDA with $\alpha = 0.55$ to SQDA and SLDA with a plastic (non-fixed) covariance matrix.

For comparison, we use the models and results of [14] as we follow the same experiment settings. We don't use models that require task labels as this is not compatible with this more general class incremental learning.

5.2. Initialization

As was done in previous works [14, 6, 36], we initialize F and G with a 100 fixed randomly selected classes. We use the same weights for G as [14] for the first 100. The 900 remaining classes are trained incrementally with a fixed representation for G as mentioned in section 4.

5.3. Results

As shown in table 5.3, our method outperforms the streaming state-of-the-art by **5 %** and is very close to the offline training of the last layer. SRDA also outperforms iCarl, and End-to-End, which are methods that update the whole model and can iterate multiple times on the data. It should be noted that SQDA, as mentioned earlier, struggles in high-dimensional settings due to the limited per-class availability of data points required for accurate estimation of covariance matrices. To address this limitation, SRDA serves as a corrective measure by leveraging the well-estimated LDA covariance matrix in combination with the estimated class covariance matrices. The figure 3 provides visual evidence supporting our findings, with a grid search CV revealing the optimal α value of 0.55 for this experiment. Better results can potentially be achieved by using a more recent backbone, such as EfficientNets [42], enabling higher accuracy with a lighter model more adapted to on device-learning.

Table 1. Ω_{all} accuracy on ImageNet. The results marked with * are taken from [14] as our experiment follows the same conditions.

Models	Streaming	CLS-IID
Output Layer Only:		
Fine-Tuning*	Yes	0.146
ExStream* [13]	Yes	0.569
SLDA [14]	Yes	0.752
SQDA (ours)	Yes	0.677
SRDA (ours)	Yes	**0.801**
Representation Learning:		
Fine-Tuning*	Yes	0.121
iCaRL* [36]	No	0.692
End-to-End* [6]	No	0.780
Offline Upper Bounds:		
Offline (Last Layer)	No	0.853
Offline	No	1.000

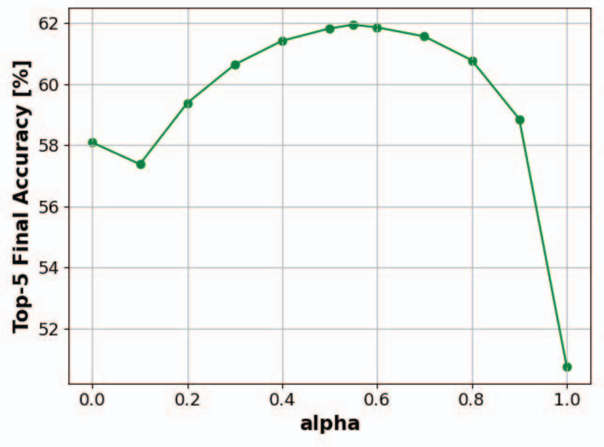

Figure 3. Variations of Top-5 Accuracy on ImageNet ILSVRC-2012 with regard to α. An $\alpha = 0$ represents a regular SLDA, whereas an $\alpha = 1$ represents an SQDA.

5.4. Hyperparameter tuning

Because this method requires the adjustment of a hyperparameter, alpha, one would think that it cannot be readily used out of the box. However, in contrast to regular machine learning hyperparameters, alpha can be modified at the end of training with minimal additional computational costs. This is due to the independent calculation of the two covariance matrices. Consequently, the model can be trained using SRDA and tuned with a quick grid search CV at the end utilizing the validation dataset without retraining. In cases where a validation dataset is unavailable, a potential solution is to maintain a small, class-balanced buffer specifically for hyperparameter tuning, which can be employed at the end of training. This enables this classification technique to be directly used and competitive with other Streaming Learning algorithms.

5.5. Computation

Due to its quadratic complexity, our algorithm takes 12 hours to compute, which is considerably higher than SLDA, which takes 30 minutes on ImageNet. Nonetheless, this is still comparatively manageable compared to other batch learning and streaming algorithms. For example, according to [14] and our experiences, ExStream takes 64 hours, and iCarl [36] 35 hours on the same hardware.

5.6. Memory usage

As it is with computational consumption 5.5, SRDA consumes more memory than SLDA as it has to store a covariance per class compared to one covariance matrix in SLDA. For instance, in the case of ImageNet ILSVRC-2012 [39] one needs $(1000 \times 4 \times (512^2 + 512))$ bytes which is equivalent to 1.051 GB. For comparison, SLDA requires 0.001

GB, ExStream requires 0.041GB, and iCarl requires 3.011 GB.

6. Conclusion & Discussions

We presented Deep Streaming Regularized Discriminant Analysis, a generative classifier able to adapt to non-iid data streams and outperform existing batch and streaming learning algorithms when paired with a CNN. We outperformed SLDA by 5%, iCarl by 11%, and End-to-End by 2%. This is an impressive result considering that both iCarl and End-to-End update the whole network and should intuitively beat a method only focusing on the last layer.

This method provides better results than SLDA at the cost of computation and memory but remains comparatively manageable compared to other methods. SQDA is better suited for low dimensional and low class counts problems, while SRDA manages to adapt to high dimensional problems with the correct regularization parameter alpha that can be found at the end of training with minimal additional computational costs

For use cases where one would like to combine the speed of SLDA and the performance of SRDA, one can imagine a model where SLDA is used for rapid learning while SRDA slowly trains in the background enabling improved accuracy in the long run.

Finally, this method represents a step forward in the research of Sustainable AI. As presented by [7], Continual Learning is a promising candidate for achieving Sustainable AI. This case of Streaming Learning justifies this choice even further as it presents a more realistic application that respects the principles of Sustainable AI, including efficiency, privacy, and robustness. Our deep SRDA has many potential applications, including robotics, edge learning, and human-machine interfaces. It removes the need to store the data as the model can learn on data streams, learning at approximately 28Hz for our experiment on ImageNet. More importantly, it enables on-device Continual Learning, removing the need for retraining and thus saving resources.

Appendix

The models were trained using these parameters:

- **Offline**: Same parameters as [14]. SGD for 90 epochs, with $lr = 0.1$ with decay at 10 30 and 60 epochs, $momentum = 0.9$ and weight decay of 10^{-4}.

- **iCarl**: Parameters from [36], and stored 20 exemplars per class.

- **ExStream**: Same parameters as offline with 20 exemplars per class.

Acknowledgments

This work was performed as part of the DREAM project. The DREAM project received funding from the European Union's European Innovation Council (EIC) under grant agreement No 101046451.

References

[1] Wickliffe C. Abraham and Anthony Robins. Memory retention – the synaptic stability versus plasticity dilemma. 28(2):73–78, 2005.

[2] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. SS-IL: Separated softmax for incremental learning.

[3] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget.

[4] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[5] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline.

[6] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. 11216:241–257, 2018. Book Title: Computer Vision – ECCV 2018 ISBN: 9783030012571 9783030012588 Place: Cham Publisher: Springer International Publishing.

[7] Andrea Cossu, Marta Ziosi, and Vincenzo Lomonaco. Sustainable artificial intelligence through continual learning, 2021.

[8] Kudithipudi et al. Biological underpinnings for lifelong learning machines. 4(3):196–210, 2022.

[9] Robert M. French. Catastrophic forgetting in connectionist networks. 3(4):128–135, 1999.

[10] Jerome H. Friedman. Regularized discriminant analysis. 84(405):165–175, 1989.

[11] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. A survey of classification methods in data streams. In Charu C. Aggarwal, editor, *Data Streams: Models and Algorithms*, Advances in Database Systems, pages 39–59. Springer US, 2007.

[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2009.

[13] Tyler L. Hayes, Nathan D. Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning.

[14] Tyler L. Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.

[16] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. pages 831–839, 2019.

[17] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 97–106. Association for Computing Machinery, 2001.

[18] Ronald Kemker and Christopher Kanan. FearNet: Brain-inspired model for incremental learning.

[19] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. 32(1), 2018. Number: 1.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. 114(13):3521–3526, 2017.

[21] Łukasz Korycki and Bartosz Krawczyk. Streaming decision trees for lifelong learning. In Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and Jose A. Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, volume 12975, pages 502–518. Springer International Publishing, 2021. Series Title: Lecture Notes in Computer Science.

[22] Cecilia S. Lee and Aaron Y. Lee. Clinical applications of continual learning machine learning. 2(6):e279–e281, 2020.

[23] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching.

[24] Jin Li, Zhong Ji, Gang Wang, Qiang Wang, and Feng Gao. Learning from students: Online contrastive distillation network for general continual learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 3215–3221. International Joint Conferences on Artificial Intelligence Organization, 2022.

[25] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3925–3934. PMLR, 2019. ISSN: 2640-3498.

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting.

[27] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. 469:28–51, 2022.

[28] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. pages 3589–3599, 2021.

[29] Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning.

[30] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification.

[31] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Gordon H. Bower, editor, *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, 1989.

[32] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. 35(11):2624–2637, 2013. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[33] J I Mwnro and M S Paterson. Selection and sorting with limited storage. 1980.

[34] Shaoning Pang, Seiichi Ozawa, and Nikola Kasabov. Incremental linear discriminant analysis for classification of data streams. 35(5):905–914, 2005.

[35] Daniel Philps, Tillman Weyde, Artur d'Avila Garcez, and Roy Batchelor. Continual learning augmented investment decisions.

[36] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning.

[37] Amanda Rios and Laurent Itti. Closed-loop memory GAN for continual learning.

[38] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning.

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge.

[40] Khadija Shaheen, Muhammad Abdullah Hanif, Osman Hasan, and Muhammad Shafique. Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks.

[41] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay.

[42] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks, 2020.

[43] D. M. Titterington, Adrian F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985. Google-Books-ID: hZ0QAQAAIAAJ.

[44] Amal Rannen Triki, Rahaf Aljundi, Mathew B. Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1329–1337, 2017.

[45] Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. 4(12):1185–1197, 2022. Number: 12 Publisher: Nature Publishing Group.

[46] Joshua T. Vogelstein. Lifelong learning forests, 2023. Section: Technical Reports.

[47] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning.

[48] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[49] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence.

[50] Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. Task agnostic continual learning using online variational bayes.