Self-supervised continual learning for object recognition in image sequences

Ruiqi Dai, Mathieu Lefort, Frédéric Armetta, Mathieu Guillermin, Stefan Duffner

Self-supervised continual learning for object recognition in image sequences

Ruiqi Dai[1,3], Mathieu Lefort[2,3], Frédéric Armetta[2,3], Mathieu Guillermin[4], and Stefan Duffner[1,3]

[1] INSA Lyon, Université de Lyon
[2] Université Claude Bernard Lyon 1, Université de Lyon
[3] LIRIS, CNRS, UMR5205
[4] Université Catholique de Lyon, France

Abstract. The autonomous learning of different objects in images, with a continual and unsupervised context, relies on detecting unknown objects and recognizing known ones based on the learned visual representation. Novelty detection is challenging because of the internal representation drifts of known objects not been seen for a long time. Most existing approaches either perform *offline* unsupervised learning on a large dataset, or continual *supervised* learning. Nevertheless, very few existing approaches propose unsupervised continual learning for object recognition. In this paper, we propose a new neural network-based approach for continually learning representations of objects from image sequences, that is able to autonomously detect novel objects and to recognize previously learned ones during training. It is based on a statistical test, performed on internal representations, adapted to counterbalance the concept drift, without storing any image. Experimental results show that our approach outperforms the state of the art on MNIST and Fashion-MNIST datasets. In particular, our approach avoids to over-segment the distribution of clusters, which artificially increases traditional indicators such as clustering accuracy.

1 Introduction

For an agent, it is very challenging to autonomously and continuously learn and make use of object representations of its open environment. The system has to detect novelty and introduce a new concept or class if necessary while maintaining the already acquired knowledge. This is part of the more general problem of finding a meaningful and robust representation under the stability-plasticity dilemma. Recently, unsupervised visual representation learning methods based on deep neural networks have been proposed [2, 11]. But learning these models online with a continuous stream of images is not straightforward due to the nature of stochastic gradient descent optimization and the fact that most of them rely on an i.i.d. data assumption which is not valid in open environments.

The literature on continual learning with neural networks [3, 17, 20] partially responds to this issue by applying strategies alleviating catastrophic forgetting, such as experience replay [3, 20], which either retain a memory of real images or generate new ones. In addition, *novelty detection* is an important aspect in

unsupervised continual learning. However, in current representation learning approaches based on generative deep neural networks, the problem of correctly re-identifying reappearing objects or detecting unknown categories is not explicitly addressed. In fact, most existing algorithms rely on the observation likelihood, the loss function or a separate classifier [3], but these methods have some serious drawbacks with continuously evolving models and are difficult to calibrate [6]. As a consequence, they tend to keep expanding ignoring the true number of categories. Moreover, more advanced techniques [7, 14] operate on statistical estimates of sample distributions, but in general they can only be applied *offline*.

In this paper, we target autonomous agent systems and propose a deep neural network model performing continual learning for visual object recognition. In this setting, the objects are learned in sequential order and are presented class by class, yet can reappear later. Our model is a modified version of a recently proposed Variational Auto-Encoder (VAE) model for continual unsupervised learning [3]. The main contribution of this paper is a new method to detect new object classes and recognize old ones in an image stream based on a statistical hypothesis test on the distributions of internal representation vectors. Our algorithm does not store any previous processed image, it only relies on the first two moments of the distributions that are estimated on-line during training.

2 Related work

Novelty detection. Detecting novelty, anomalies or outliers in a data sample distribution has been studied for several decades [9, 19]. Some of the works consider the problem as one-class classification problem, i. e. only modelling the nominal class, or by creating a single artificial class, but their performances suffer when the number of categories increases [6]. In classification applications, one usual approach is to infer abnormality from the output scores of the classifier. However, samples from unknown classes may produce high confidence values by strongly activating one of the known classes [6, 14]. For example, in neural networks, the softmax function indicates the confidence in the classification, but the interpretation as a true probability estimate is questionable [8]. Multi-class novelty detection can be formulated as "open-set recognition" [6], i.e. examples of unknown classes are explicitly handled by the model and rejected. For neural networks, for instance, models based on the calibration (of softmax) have been proposed such as ODIN [15] or G-OpenMax [5]. Also, removing the softmax normalisation may improve the discrimination between inliers and outliers [16]. A more probabilistic approach consists in using the likelihood ratio between inliers and a background distribution for outlier detection [21, 23]. Other methods compute the uncertainty of model predictions using ensembles of deep models [13].

Despite these advances, two major challenges with generative neural networks remain: it may be difficult to disentangle known from unknown classes [6, 18], and most methods operate offline, which either requires all the feature vectors extracted for known classes or explicitly train a separate mapping function that is independent from the learnt representation [1].

Continual learning with neural networks. Various methods have been proposed to alleviate catastrophic forgetting, such as *regularization strategies* [12,25] that try to conserve weights w.r.t. their relevance to already learned classes. Other strategies include *structural approaches* [17, 22] that dynamically expand network structures for new tasks, or *experience replay* [20] with either partial storage of real training examples or generated examples for learned categories [3]. However, these approaches are usually applied to supervised settings.

Among the unsupervised approaches, the Self-Taught Associative Memory (STAM) architecture [24] uses a specific hierarchical feature representation based on image patch exemplars of different sizes obtained from clustering, which may limit its scalability. SOINN [4] proposed a model with Self-Organizing Incremental Neural Network based on a distance metric to incrementally learn the topology of input data. Our model is based on the Continual Unsupervised Representation Learning (CURL) model [3] that will be presented in section 3.

3 Representation learning algorithm and model

Our model is an extension of [3], a type of Variational Auto-Encoder (VAE) with a Gaussian Mixture Model (GMM), adapted to a class-incremental learning setting. More formally, it estimates the probability $p(x, y, z) = p(y)p(z|y)p(x|z)$ of the input x, the label y and the latent code z, using variational inference, and approximates the posterior $p(y, z|x)$ with $q(y, z|x) = q(y|x)q(z|x, y)$, where $q(y|x)$ is the output of a dense layer followed by softmax determining the component posterior given an input x, and $q(z|x, y)$ is the distribution of z encoded by the component-wise encoder. Instead of a single multivariate Gaussian as in classical VAE, CURL uses several components modelled as dense neural layers that output the different means and variances, and that are dynamically added during training when a certain number of outliers are detected. A modified ELBO (Evidence Lower Bound) objective is optimized during learning: $E(x) =$

$$\sum_{k=1}^{K} q(y=k|x)\Big[\underbrace{\log p(x|\widetilde{z}^{(k)})}_{\substack{\text{component-wise}\\ \text{reconstruction}}} - \underbrace{\mathrm{KL}(q(z|x,y=k)||p(z|y=k))}_{\substack{\text{Kullback-Leibler divergence}\\ \text{regularization on z}}}\Big] - \underbrace{\mathrm{KL}(q(y|x)||p(y))}_{\substack{\text{categorical}\\ \text{regularization}}} \tag{1}$$

with $\widetilde{z}^{(k)} \sim q(z|x, y = k)$ being the sampled latent code, $p(z|y = k)$ and $p(y)$ being the prior distributions (normal and uniform respectively). By maximizing Eq. 1, the model maximizes the data likelihood via reconstruction (first term), and regularises the model such that z tends to be normally distributed (second term) and samples are evenly distributed over components (third term).

During training, the model alleviates catastrophic forgetting by following the *mixture generative replay* strategy, mixing real examples of the current category with generated images of past categories. To detect outliers, the model uses the ELBO loss (Eq. 1) as an indicator of learning quality, modelling the likelihood of an example belonging to a learned category. For more details, refer to [3].

The authors also introduced a supervised form of the ELBO loss that replaces the unsupervised ELBO objective (Eq. 1) when used in a supervised setting (in this case the component y to update is not determined by the internal classifier $q(y|x)$ but selected by y_m): $E_{sup}(x) =$

$$\log q(y = y_m|x) + \log p(x|\widetilde{z}^{y_m}, y = y_m) - KL(q(z|x, y = y_m)||p(z|y = y_m)) \quad (2)$$

In our model the labels y_m are a self-supervision signal generated automatically based on a statistical hypothesis test - a two-sample t^2 Hotelling test that we adapted (cf. section 4.1) - performed on the internal representation of the model.

4 Proposed Approach

4.1 Continual detection and recognition of objects

In our model, one object category is supposed to be modelled by a *single* component, contrary to CURL that uses a GMM model allowing multiple components per category. As the latent variable z for each category tends to follow a multivariate normal distribution, because of the KL regularisation term during training, to decide if the current observation batch corresponds to a given class, we perform a two-sample Hotelling t^2 test [10], which is a statistical test for multivariate normal distributions. To this end, we compute the t^2 statistics:

$$t^2 = \frac{n_y * n_b}{n_y + n_b}(\bar{z}_y - \bar{z}_b)^T \widehat{\Sigma}^{-1}(\bar{z}_y - \bar{z}_b) \ , \quad (3)$$

with $\widehat{\Sigma}$ being the pooled covariance matrix determined by

$$\widehat{\Sigma} = \frac{(n_y - 1)\widehat{\Sigma}_{y_{sh}} + (n_b - 1)\widehat{\Sigma}_b}{n_y + n_b - 2} \ , \quad (4)$$

$\bar{z}_y$ and $\Sigma_{y_{sh}}$ the sample mean and covariance matrix of latent variable z corresponding to an object category y, $\bar{z}_b$ and Σ_b the sample mean and covariance matrix of the input batch and n_y, n_b the sample sizes of the two distributions respectively. The t^2 distribution follows the F distribution up to a factor, where d is the dimension of z:

$$\frac{n_y + n_b - d - 1}{(n_y + n_b - 2)d} t^2 \sim F(d, n_y + n_b - 1 - d) \quad (5)$$

Our null hypothesis H_0 is that the two means μ_y, μ_b of object class y and input batch b are equal, and we reject it if the left hand side of Eq. 5 is below a critical value related to a given p-value threshold. In practice, we compute the p-values for the tests between the current batch and each trained object category, and we select the class with the highest p-value if it is above the defined threshold. Otherwise, it is considered belonging to a new object category.

4.2 Online parameter estimation

In our continual learning setting, the embedding in the latent space of the VAE gradually evolves. To compute the mean and covariance of previous objects without storing past images, we approximate them by running averages. However, these approximated covariance matrices slightly underestimate the actual variance. This may lead to too large values in the t^2 statistics (Eq. 3) and eventually to very small p-values for known classes. To alleviate this problem, we apply a shrinkage operation to the running covariance matrices such that the diagonal entries are more homogeneous and the difference between eigenvalues is reduced.

$$\bar{z}_y(t) = (1-\alpha)\bar{z}_y(t-1) + \alpha z_y(t) \ , \tag{6}$$

$$\widehat{\Sigma}_y(t) = (1-\alpha)\widehat{\Sigma}_y(t-1) + \alpha(z_y(t) - \bar{z}_y(t))^T(z_y(t) - \bar{z}_y(t)) \ . \tag{7}$$

$$\widehat{\Sigma}_{y_{sh}}(t) = (1-\gamma)\widehat{\Sigma}_y + \gamma * \frac{\mathrm{tr}(\widehat{\Sigma}_y)}{d} I \ , \tag{8}$$

where $z_y(t)$ is the embedding of class y, $\alpha \in (0,1)$ a small update factor, $\gamma > 0$ is the shrinkage coefficient and I the identity matrix. The equations are applied for the current class with real observations and for all the other classes with synthetic examples from generative replay. Finally, we do the Hotelling t^2 test (Eq. 4) using $\bar{z}_y$ and $\widehat{\Sigma}_{y_{sh}}$.

5 Experiments

5.1 Protocol

We evaluated our proposed approach on MNIST and Fashion-MNIST, each including 10 classes, a total of 60000 images of size 28x28 for training and 10000 images for test. The VAE architecture we used is the same as the one of CURL, i. e. a 4-layer MLP {1200, 600, 300, 150} as encoder and a two-layer MLP {500, 500} as decoder, with a 32-dimensional latent space. The learning rate is set to 10^{-3} with an Adam optimizer. The size of the outlier buffer is 100 for CURL, 200 for our model and for "CURL with HT" (as the test is performed on batches, we slightly increase the buffer to contain 2 batches and avoid potential fluctuations). The batch size is 100, thus $n_b = 100$ and n_y is empirically set to 20.

Our protocol consists of two consecutive sequences. In the first one, we present half of the training examples class by class in a random order to test the ability of the model to detect new classes. In the second one, we present the second half of the training data still class by class, in the same order as in the first sequence, to test the recognition performance of the model. We compared our approach with CURL [3], SOINN [4] and CURL combined with Hotelling t-squared test as an ablation study.

model	AMI	ARI	# components	accuracy
CURL [3]	0.518 ± 0.013	0.156 ± 0.02	126.6 ± 17.74	1.0 ± 0.0
CURL with HT	0.6 ± 0.04	0.41 ± 0.025	22.6 ± 1.69	0.87 ± 0.04
SOINN [4]	0.367 ± 0.002	0.013 ± 0.008	1507 ± 11.34	1.0 ± 0.0
Ours	$\mathbf{0.778 \pm 0.012}$	$\mathbf{0.769 \pm 0.02}$	$\mathbf{11 \pm 1.41}$	$\mathbf{1.0 \pm 0.0}$

Table 1. Comparison with the state of the art on MNIST (mean of 3 runs ±SD).

model	AMI	ARI	# components	accuracy
CURL [3]	0.429 ± 0.004	0.1006 ± 0.0111	170.0 ± 0.0	0.993 ± 0.004
CURL with HT	0.473 ± 0.0054	0.25 ± 0.008	31.3 ± 6.34	0.857 ± 0.03
SOINN [4]	0.342 ± 0.0016	0.016 ± 0.0003	1009 ± 17.518	$\mathbf{1.0 \pm 0.0}$
Ours	$\mathbf{0.57 \pm 0.02}$	$\mathbf{0.395 \pm 0.03}$	$\mathbf{13.33 \pm 0.94}$	0.798 ± 0.002

Table 2. Comparison with the state of the art on Fashion-MNIST (mean of 3 runs ± SD).

5.2 Results

To evaluate the quality of the learned clustering, we compute different standard metrics on the two test sets: accuracy (the label of each component is obtained by post labelling via majority vote), the Adjusted Mutual Information (AMI) and the Adjusted Random Index (ARI) measuring the correspondence of the learned clustering w.r.t. the Ground Truth. Our model considers that all examples of a batch belong to the category, while the other tested models evaluate each example individually. For a fair comparison, the other models are also evaluated batch by batch via majority vote of prediction amongst the batch.

On the MNIST dataset (table 1), our model performs better than all other models for all indicators. In particular, the number of components is much closer to the real number of categories, thus greatly improving the AMI and ARI score, while not impacting the accuracy which remains at 100%. It is interesting to note that the application of the Hotelling test on CURL, decreases significantly the number of learned components. However, this does allow the model to reach the AMI and ARI score of our model and induces a drop in accuracy performance. This validates the need of using the supervised ELBO loss in our model.

On the Fashion-MNIST dataset (table 2), the trend for AMI and ARI are similar to the ones observed on MNIST. However, here, this is obtained at the cost of a drop in accuracy. This may be explained as the Fashion-MNIST dataset is harder and by the fact that having a higher number of clusters facilitates a high accuracy (as the chance of mixing different classes in one component is reduced). However, for an autonomous agent, we prefer to have a smaller number of clusters as less examples can be required to label them.

In figure 1 we illustrate the evolution of model performance during training on Fashion-MNIST on the *detection accuracy*, i. e. the binary classification of "known" and "unknown" classes, and the *clustering accuracy* of the classes that have been learned by the agent, i. e. known classes (except for those considered by error as unknown). We can observe that each new class is detected at the right time and that the model is not subject to catastrophic forgetting.

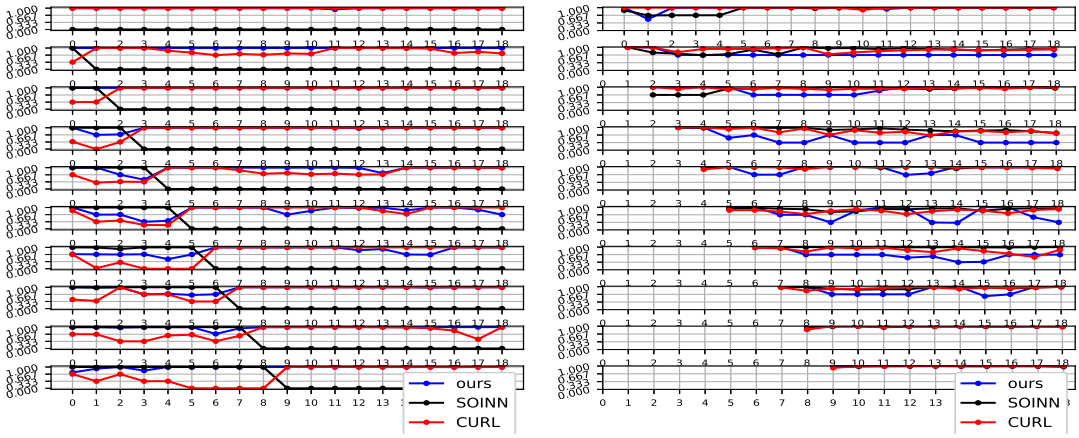

Fig. 1. The evolution of detection (left) and clustering accuracy (right) on Fashion-MNIST during training. The rows represent the 10 classes in the learned order.

6 Conclusion

In this paper, we proposed a self-supervised continual object representation learning algorithm extending the CURL model [3]. Our contribution lies in combining the model with a statistical hypothesis test allowing to detect unknown categories and to recognize previously learned categories, and in this way, to self-supervise the learning process without storing any previous examples. Compared to the state of the art, our model demonstrates its capacity to recognize learned objects in an online scenario while avoiding creating and reallocating new components for learned categories. Thus the model becomes more effective in automatically detecting the number of categories. Our proposal permits to moderate the amount of over-segmentation and achieves better performance in terms of standard clustering metrics AMI and ARI compared to the state-of-the-art algorithms CURL and SOINN [4] but may have lower accuracy due to fewer component creation. The introduced statistical test allows to detect properly the novel classes and recognize learned categories. In the future, we will evaluate our approach on more realistic scenarios for autonomous agents with video streams of more complex objects in difficult and varying environments. The balance to be found between the quantity of over segmentation and accuracy can be of primary interest in this context as well.

References

1. P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler. Kernel null space methods for novelty detection. In *CVPR*, 2013.
2. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020.

3. F. Dushyant, R.and Visin, A. Rusu, R. Pascanu, Y. W. Teh, and R. Hadsell. Continual unsupervised representation learning. In *NeurIPS*, 2019.
4. S. Furao and O. Hasegawa. An incremental network for on-line unsupervised classification and topology learning. *Neural networks*, 19(1):90–106, 2006.
5. Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi. Generative openmax for multi-class open set classification. In *BMVC*, 2017.
6. C. Geng, S.-J. Huang, and S. Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.
7. D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.
8. S. Hess, W. Duivesteijn, and D. Mocanu. Softmax-based classification is k-means clustering: Formal proof, consequences for adversarial attacks, and improvement through centroid based tailoring. *CoRR*, abs/2001.01987, 2020.
9. V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126, 2004.
10. H. Hotelling. The generalization of student's ratio. *Annals of Mathematical Statistics*, 2(3):360–378, 1931.
11. D.P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.
12. J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A Rusu, K. Milan, J. Quan, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
13. B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, pages 6402–6413, 2017.
14. K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.
15. S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.
16. W. Liu, X. Wang, J. D. Owens, and Y. Li. Energy-based out-of-distribution detection. In *Advances in neural information processing systems*, 2020.
17. D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.
18. E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *ICLR*, 2019.
19. Ma. A.F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.
20. S. Rebuffi, A. Kolesnikov, G. Sperl, and C. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017.
21. J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2019.
22. A. Rusu, N. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
23. J. Serrà, D. Alvarez, V. Gómez, O. Slizovskaia, J. F. Núnez, and J. Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2020.
24. J. Smith, C. Taylor, S. Baer, and C. Dovrolis. Unsupervised progressive learning and the STAM architecture. In *IJCAI*, 2021.
25. F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995, 2017.