A SIMPLE, FAST ALGORITHM FOR CONTINUAL LEARNING FROM HIGH-DIMENSIONAL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

As an alternative to resource-intensive deep learning approaches to continual learning, we propose a simple, fast algorithm inspired by adaptive resonance theory (ART) and coupled with incremental principal component analysis (IPCA). IPCA is used to avoid the curse of dimensionality and improve the performance of ART's category matching phase. Our algorithm is motivated by the observation that changes to the input representation due to IPCA updates can be reflected in the model by simply applying the current version of IPCA to the model's previously learned weights.

1 INTRODUCTION

In the continual learning problem setting, we wish to efficiently learn a sequence of tasks without forgetting how to perform previously learned tasks. Many approaches have been proposed to address this problem, with recent work focused mainly on deep learning (Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017; Serrà et al., 2018). While these approaches boast impressive predictive power, they are limited by a host of drawbacks including high computational costs, large memory requirements, and poor performance on nontrivial continual learning benchmarks with long task sequences and task-identification information withheld (Farquhar & Gal, 2018; Díaz-Rodríguez et al., 2018).

We propose an alternative, matching-based algorithm which combines ideas from ART with IPCA to improve predictive performance for high-dimensional data. ART is an attractive solution to continual learning because of its proven stability guarantees which imply that catastrophic forgetting will not occur with respect to the training data (Grossberg, 2020). PCA is a well-known linear compression technique which can be effectively used as a data preprocessing step for both supervised and unsupervised learning tasks. Many incremental versions of PCA have been proposed to efficiently update principal component estimates without recomputing eigenvalues from the full covariance matrix (Weng et al., 2003; Zhao et al., 2006; Ross et al., 2008).

2 PROPOSED MODEL

We begin by formally defining our notation and problem statement. We are given a set of tasks $\mathcal{T}_1 \dots \mathcal{T}_T$ where each task $\mathcal{T}_i$ has a corresponding dataset $D_i = (\boldsymbol{X}_i, \boldsymbol{y}_i)$. We focus on supervised classification tasks with features $\boldsymbol{X}_i \in \mathbb{R}^{n_i \times d}$ and categorical labels $\boldsymbol{y}_i \in \mathbb{N}^{n_i \times 1}$, though our proposed model can easily be extended to unsupervised clustering. We reduce our data's dimension from $d \rightarrow k$ using IPCA matrix $\boldsymbol{P} \in \mathbb{R}^{k \times d}$ whose rows consist of the top-k estimated eigenvectors of the data's covariance matrix. The weights in the original space and IPCA reduced space are denoted by $\boldsymbol{w}_{org} \in \mathbb{R}^{d \times 1}$ and $\boldsymbol{w}_{red} \in \mathbb{R}^{k \times 1}$ respectively, with $\boldsymbol{w}_{red}^i = \boldsymbol{P}_i \boldsymbol{w}_{org}$ denoting the weight in reduced space under iteration i of IPCA.

Our proposed model does not require task-identifiers, though this information may be optionally included to improve performance. It is applicable to class-incremental, task-incremental, and domain-incremental learning, and only requires that features have consistent dimensionality across all tasks. Algorithm 1 describes the training procedure while the inference procedure and intuition behind the model are discussed in Appendix A.1. Time and space complexity are given in Appendix A.2.

When applying IPCA as a preprocessing step in the continual learning setting, a key question emerges. When data from a new task arrives, the IPCA matrix is updated, resulting in a new rep-

Algorithm 1 Supervised training in batch mode

1: **Input:** Set of tasks $\mathcal{T}_1 \dots \mathcal{T}_T$, each task $\mathcal{T}_i$ has its own set of training data $D_i = (\boldsymbol{X}_i, \boldsymbol{y}_i)$, pairwise similarity function $S(\cdot, \cdot)$, vigilance ρ, reduced dimension k, learning rate β
2: $W = \emptyset$ ▷ Weights in original space
3: $C = \emptyset$ ▷ Map from weights to labels
4: **for** each task $\mathcal{T}_i$ **do**
5: Update $\boldsymbol{P}_{i-1}$ with new data $\boldsymbol{X}_i$ to get $\boldsymbol{P}_i$
6: **for** each training example $(\boldsymbol{x}_j, y_j) \in (\boldsymbol{X}_i, \boldsymbol{y}_i)$ **do**
7: $\boldsymbol{w}^*_{org} = \underset{\boldsymbol{w}_{org} \in W}{\operatorname{argmax}}\ S(\boldsymbol{P}_i \boldsymbol{x}_j, \boldsymbol{P}_i \boldsymbol{w}_{org})$ ▷ Similarity in reduced space
8: **if** $S(\boldsymbol{P}_i \boldsymbol{x}_j, \boldsymbol{P}_i \boldsymbol{w}^*_{org}) \geq \rho$ and $C[\boldsymbol{w}^*_{org}] = y_j$ **then**
9: $\boldsymbol{w}^*_{org} = \beta \boldsymbol{x}_j + (1 - \beta) \boldsymbol{w}^*_{org}$ ▷ Learn weights in original space
10: **else**
11: $W = W \cup \boldsymbol{x}_j$ ▷ Set new weight to current sample
12: $C[\boldsymbol{x}_j] = y_j$ ▷ Map new weight to correct label
13: **end if**
14: **end for**
15: **end for**

resentation of previously learned data. *How should the model's previously learned parameters be adjusted given the new representation of the training data in order to avoid catastrophic forgetting?*

We answer this question by observing that the updated version of IPCA can be applied to transform the model's weights (learned in the original space) since the weights are simply a linear combination of training samples. In other words, learning in the current IPCA reduced space results in the same weights as previously learning in the original space, then applying the current version of IPCA. This argument is formalized in Proposition 1 with proof and further explanation in Appendix A.3.

Proposition 1. *Let $\boldsymbol{x}_1 \dots \boldsymbol{x}_m$ be the training samples which have matched to weight $\boldsymbol{w}$ under some previous IPCA transformation(s) $\boldsymbol{P}_1 \dots \boldsymbol{P}_{i-1}$ [1]. Assuming that these samples $\boldsymbol{x}_1 \dots \boldsymbol{x}_m$ still match to weight $\boldsymbol{w}$ under the current IPCA transformation $\boldsymbol{P}_i$, then learning the weight $\boldsymbol{w}^i_{red}$ in the reduced space is equivalent to first learning the weight $\boldsymbol{w}_{org}$ in the original space, then applying IPCA $\boldsymbol{P}_i$.*

3 Results

To evaluate our proposed algorithm, we perform experiments on two commonly used continual learning benchmarks – Split MNIST and Split Fashion-MNIST – in both single-head and multi-head settings. We compare our proposed IPCA model against two baselines: our model learned without any form of PCA, and our model learned on data transformed by PCA in the static (i.e. non-incremental) setting. Assuming that PCA improves matching performance, we expect the "No PCA" and "Static PCA" models to be approximate lower and upper bounds for "IPCA" model performance.

In the single-head setting, we find that our proposed model with IPCA significantly outperforms the model learned without PCA and does not significantly differ from the model learned with static PCA (using paired t-tests with significance level $\alpha = 0.05$). In the easier multi-head setting, we find that all models offer strong performance. See Table 1 for results and Appendix A.5 for further details.

Table 1: Mean accuracy and standard deviation over 25 trials. The training data in each task is randomly shuffled resulting in different presentation orderings across different trials.

	Single-Head		Multi-Head	
	MNIST	Fashion-MNIST	MNIST	Fashion-MNIST
No PCA	92.99 ± 0.41	76.09 ± 0.50	99.47 ± 0.05	98.86 ± 0.09
IPCA	94.03 ± 0.26	77.56 ± 0.60	99.46 ± 0.07	98.86 ± 0.10
Static PCA	94.04 ± 0.24	77.48 ± 0.39	99.41 ± 0.06	98.79 ± 0.13

[1] The phrase "sample $\boldsymbol{x}$ matches to weight $\boldsymbol{w}$" means that $\boldsymbol{w}$ is the weight with the maximum similarity score with respect to $\boldsymbol{x}$ in line 7 of Algorithm 1. Weight $\boldsymbol{w}$ is also referred to as the "winner" of ART's competition.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2023 Tiny Papers Track.

REFERENCES

Georgios C Anagnostopoulos and Michael Georgiopoulos. Putting the utility of match tracking in fuzzy artmap training to the test. In *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 1–6. Springer, 2003.

Gail A Carpenter, Stephen Grossberg, Natalya Markuzon, John H Reynolds, David B Rosen, et al. Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on neural networks*, 3(5):698–713, 1992.

Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: new metrics for continual learning, 2018. URL https://arxiv.org/abs/1810.13166.

Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning, 2018. URL https://arxiv.org/abs/1805.09733.

Stephen Grossberg. A path toward explainable AI and autonomous adaptive intelligence: Deep learning, adaptive resonance, and models of perception, emotion, and action. *Frontiers in Neurorobotics*, 14, 2020. ISSN 1662-5218. doi: 10.3389/fnbot.2020.00036. URL https://www.frontiersin.org/articles/10.3389/fnbot.2020.00036.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical study on solving continual learning. *arXiv preprint arXiv:2211.02633*, 2022.

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. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, mar 2017. doi: 10.1073/pnas.1611835114.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/f87522788a2be2d171666752f97ddebb-Paper.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

David Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1r3):125r141, 2008.

Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task, 2018. URL https://arxiv.org/abs/1801.01423.

Juyang Weng, Yilu Zhang, and Wey-Shiuan Hwang. Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8): 1034–1040, 2003. doi: 10.1109/TPAMI.2003.1217609.

Haitao Zhao, Pong Chi Yuen, and James T Kwok. A novel incremental principal component analysis and its application for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(4):873–886, 2006.

A APPENDIX

A.1 ALGORITHM DETAILS AND INTUITION

A verbal description of Algorithm 1 is as follows. Each time a new task arrives, the IPCA matrix is updated using all of the task's data. Sequentially, for each training sample in the task, the best-matching weight is computed using a similarity metric in the IPCA reduced space of dimension k. If the best-matching weight is sufficiently similar to the sample and it corresponds to the correct class, then the weight is updated in the original space of dimension d. Otherwise, a new weight is initialized to the current sample in the original space, with its corresponding class label recorded. At inference, simply compute similarity scores between the test sample and all learned weights in the reduced space, then predict the class corresponding to the weight with the largest similarity score.

The main idea behind our proposed algorithm is to learn weights in the original space and perform matching in the IPCA reduced space. Storing the weights in the original space allows the model to adapt to changes in the input representation caused by IPCA updates. This adaptation is done by simply applying the current version of IPCA to the previously learned weights. This idea could conceivably be extended to other types of incremental learning algorithms beyond IPCA and ART.

A.2 TIME AND SPACE COMPLEXITY

The space complexity of our model during training is $\mathcal{O}(kd + Wd)$ where W is the total number of learned weights. During training, both the current IPCA matrix and weights in the original space must be stored. The time complexity for training is $\mathcal{O}(Nkd + TWkd + NWk)$ given N total training samples and T tasks. This consists of the time required for learning/applying IPCA updates to the data, applying IPCA updates to the weights, and performing matching.

The space complexity of our trained model is $\mathcal{O}(kd + Wk)$. The time complexity for inference on a single sample is also $\mathcal{O}(kd + Wk)$. This is because the IPCA matrix and learned weights must be stored in order to transform new data and compute matches at inference. These bounds assume that the learned weights are stored in the reduced space of dimension k after all training is complete – this is a small implementation detail included in the time complexity given above for training. Note that the number of weights W can be controlled by setting the vigilance parameter ρ, with smaller values of ρ resulting in fewer weights and coarser granularity.

A.3 APPROXIMATION OF STATIC PCA

Proposition 1 attempts to establish an equivalence between learning incrementally in the original space and learning statically in the reduced space. If Proposition 1 holds for all weights $w \in W$ and for all iterations $i = 1 \dots T$, then the weights learned in the IPCA reduced space will be equivalent to those learned in the static PCA reduced space, modulo the approximation error incurred by IPCA. Here, "static PCA" refers to first learning and applying PCA to all of the training data, then learning the weights solely in the PCA reduced space.

Proof of Proposition 1. This equivalence can be easily shown using linearity, with coefficients α determined by the learning rate:

$$\begin{aligned} \boldsymbol{w}_{red}^{i} &= \alpha_1 \boldsymbol{P}_i \boldsymbol{x}_1 + \ldots + \alpha_m \boldsymbol{P}_i \boldsymbol{x}_m \\ &= \boldsymbol{P}_i(\alpha_1 \boldsymbol{x}_1 + \ldots + \alpha_m \boldsymbol{x}_m) \\ &= \boldsymbol{P}_i \boldsymbol{w}_{org} \end{aligned} \tag{1}$$

The top line in 1 represents learning in the current IPCA reduced space, while the bottom two lines represent previously learning in the original space, then applying the current version of IPCA. □

In practice, it is unlikely that Proposition 1 will hold due to its strong assumption that matches will be the same across various iterations of IPCA. However, we do expect matches to be similar (though not exactly the same) across iterations of IPCA. Our experimental results in Table 1 demonstrate that this approximate equivalence holds in practice.

A.4 RELATION TO ART

Our proposed algorithm is inspired by ART, and essentially performs incremental winner-take-all clustering. However, there are several differences between our algorithm and ART-based algorithms. Our algorithm differs from ART as it (1) does not include distinct bottom-up and top-down matching phases, (2) uses linear rather than fuzzy intersection weight updates, and (3) avoids match tracking. Specifically, our algorithm uses the alternative to match tracking described in Anagnostopoulos & Georgiopoulos (2003) which satisfies Fuzzy ARTMAP's incremental learning principle.

Because of these differences, some of ART's properties do not apply to our proposed algorithm. For example, the use of fuzzy intersection weight updates ensures that categories only shrink, implying that weights do not repeat previously held values (Carpenter et al., 1992). (We do not use fuzzy intersection as it is nonlinear, thus incompatible with Proposition 1). Still, our algorithm retains arguably the most important property of ART: the use of competition to protect previously learned knowledge. Understanding which properties are necessary in order to avoid catastrophic forgetting while achieving strong generalization performance is an active area of research (Kim et al., 2022).

A.5 EXPERIMENTAL DETAILS

In our experiments in Section 3, we use 2000 train samples and 500 test samples of each class. We use PCA/IPCA to reduce to dimensions $k = 200$ for the MNIST dataset and $k = 250$ for the Fashion-MNIST dataset, which corresponds to preserving slightly more than 95% of the variance in the training data. Both datasets have original dimension $d = 784$. We use vigilance parameter $\rho = 0.5$, cosine similarity function $S(\cdot, \cdot)$, and adaptive learning rate $1/n_s$ where n_s denotes the total number of samples which have matched with the winning weight (i.e. the weights are simply an average of their corresponding samples). These values were determined through basic hyperparameter tuning on a validation set with 500 samples per class. We use the scikit-learn (Pedregosa et al., 2011) implementation of IPCA based on Ross et al. (2008). A link to the GitHub repository with our proposed algorithm's implementation will be included upon acceptance.

The Split-MNIST and Split-Fashion-MNIST datasets each have 10 classes, divided into 5 binary classification problems (classify 0 vs 1, 2 vs 3, 4 vs 5, 6 vs 7, 8 vs 9). Results in Table 1 are reported as the average performance across each of the 5 tasks after learning all tasks sequentially (i.e. tasks $1 - 5$ are first learned in order, then task 1 is evaluated, then task 2 is evaluated, etc). In the multi-head setting, each test task is a binary classification problem in the same form as the training tasks, and task-identifiers are provided at test time. In the single-head setting, each test task is a multiclass classification problem and task identifiers are not provided. As observed in Farquhar & Gal (2018) the multi-head setting is considerably easier than the single-head setting, and can be solved trivially by training isolated models for each task. Since the multi-head setting is commonly used in continual learning evaluations, we include it in our results for reference.