Baylor College of Medicine

UNIVERSITY OF CAMBRIDGE

RICE

Class-incremental learning with generative classifiers

Gido van de Ven, Zhe Li & Andreas Tolias

For full details:

van de Ven GM, Li Z, Tolias AS (2021) Class-Incremental Learning with Generative Classifiers. *ArXiv preprint:* arXiv:2104.10093.

Code: https://github.com/GMvandeVen/class-incremental-learning

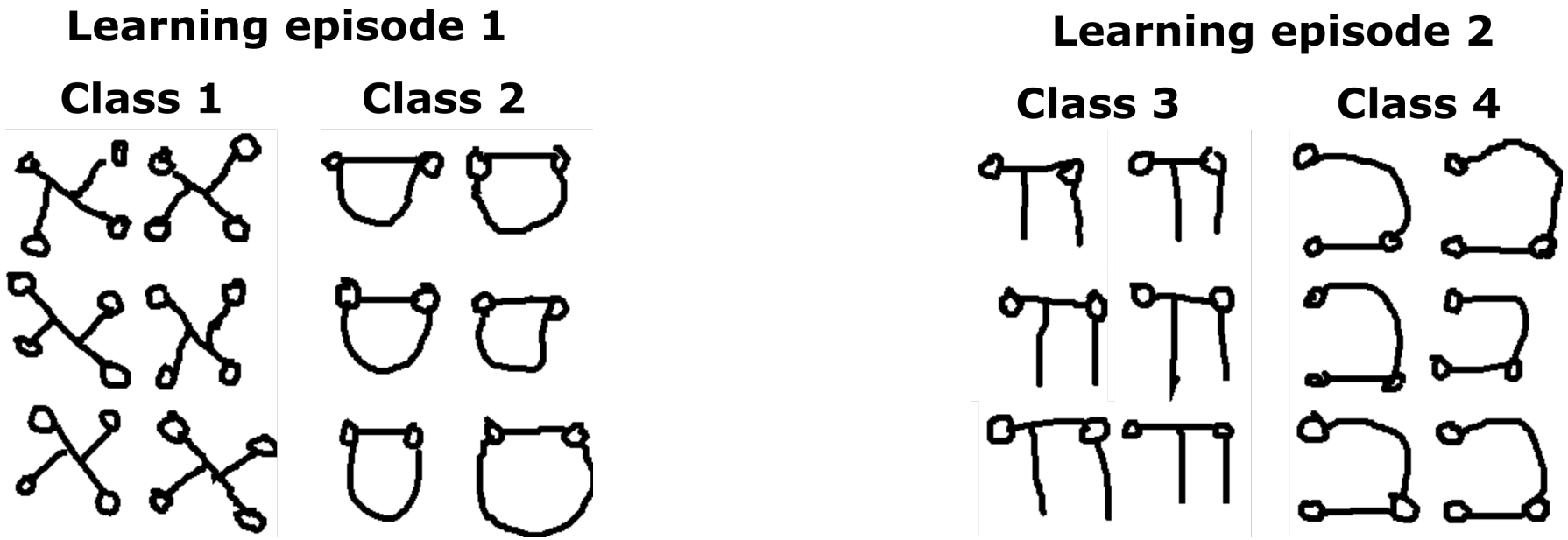
Three types of continual learning

- Task-incremental learning
  - Incrementally learn a set of clearly distinct tasks
- Domain-incremental learning
  - Learn the same type of task, but with changing contexts
- Class-incremental learning
  - Incrementally learn to distinguish between a growing number of classes

van de Ven & Tolias (2018) *NeurIPS workshop*, https://arxiv.org/abs/1904.07734

Class-incremental learning

- Main challenge:
  → Learn to distinguish between classes that are not observed together



Strategies for class-incremental learning

- ***Store some of the past data***

→ Only methods that do not store data are considered

- ***Generative replay***
  - Learn a generative model to generate samples representative of past data

→ Deep Generative Replay (**DGR**; Shin et al., 2017 *NeurIPS*)
Brain-Inspired Replay (**BI-R**; van de Ven et al., 2020 *Nature Communications*)

- ***Parameter Regularization***
  - Encourage parameters important for past tasks not to change too much when learning new tasks

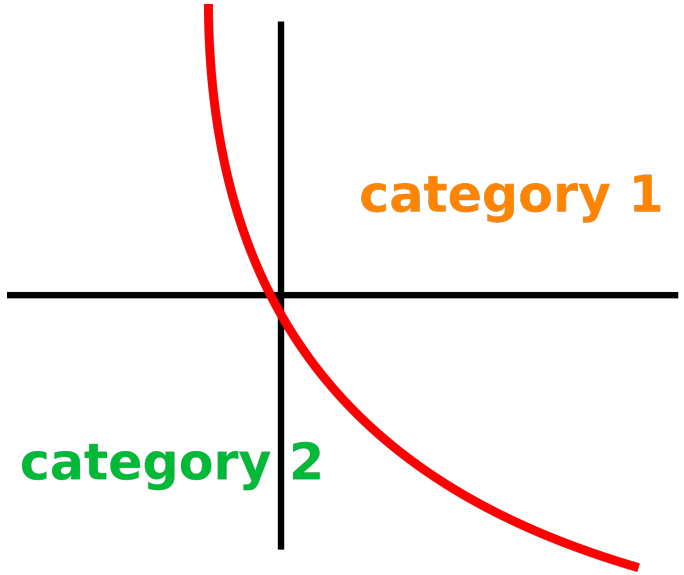
→ Elastic Weight Consolidation (**EWC**; Kirckpatrick et al., 2017 *PNAS*)
Synaptic Intelligence (**SI**; Zenke et al., 2017 *ICML*)

- ***Bias-correction***
  - Correct the bias of the output layer, which tends to only predict recently seen classes, by making the magnitude of the output weights of all classes comparable

→ CopyWeights with Re-init (**CWR**; Lomonaco & Maltoni, 2017 *CoRL*)
AR1 (Maltoni & Lomonaco, 2019 *Neural Networks*)
"**Labels trick**" (Zeno et al., 2019 *arXiv*)

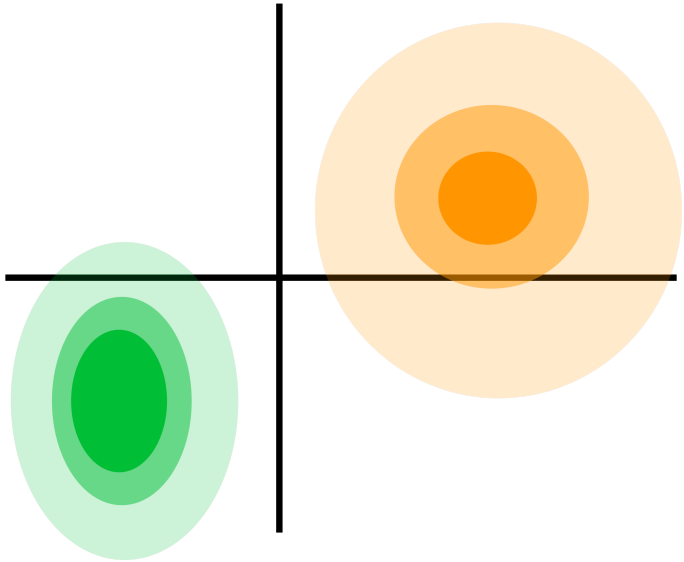
Proposed strategy: generative classification

Discriminative classifiers



- Learn rules / shortcuts / features to distinguish between the classes to be learned
- Comparison between classes is during *training*

Generative classifiers



- Learn a model / template / representation for each class to be learned
- Comparison between classes is during *inference*

Proposed strategy: generative classification

Discriminative classifiers

- Discriminative classifiers directly learn $p(y|\boldsymbol{x})$, or $\underset{y}{\operatorname{argmax}}\, p(y|\boldsymbol{x})$.
- With class-incremental learning, this is problematic because, based on the most recently seen data, the empirical version of $p(y|\boldsymbol{x})$ is heavily biased towards the newer classes

Generative classifiers

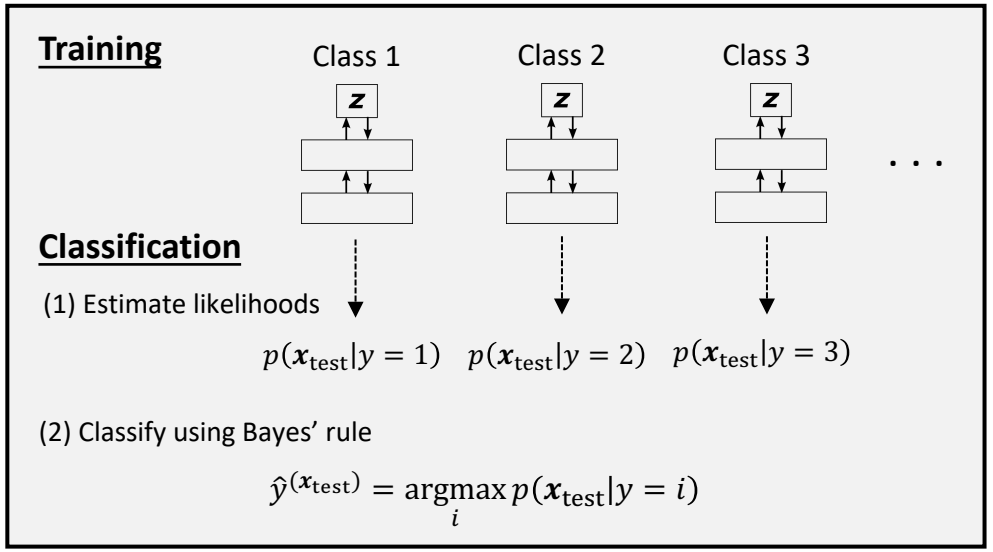
- We instead learn $p(\boldsymbol{x}, y)$, factorized as $p(\boldsymbol{x}|y)p(y)$, and classify using Bayes' rule: $p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}|y)p(y)$.
- With class-incremental learning, the empirical version of $p(\boldsymbol{x}|y)$ does not have any bias, while learning $p(y)$ without forgetting is typically straight-forward

Generative classification ***rephrases a class-incremental problem as a task-incremental problem***, whereby each 'task' is to learn a class-conditional generative model.

Implementation for a *proof-of-principle*: VAE per class & importance sampling

- To learn the distributions $p(\boldsymbol{x}|y)$, we train a separate VAE model for each class y

- When classifying a test sample $\boldsymbol{x}_{\text{test}}$, for each class y, the class-conditional likelihood $p(\boldsymbol{x}_{\text{test}}|y)$ is estimated using importance sampling



- The VAE models are chosen so that the *total* number of parameters is similar to the number of parameters used by generative replay

Class-incremental learning benchmarks

- Task-based (left) *vs*. task-free (right)

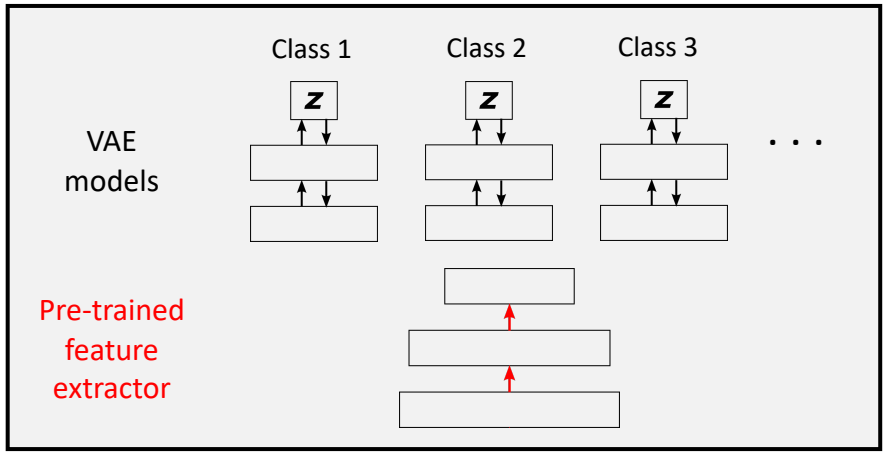
- Other important differences between studies:
  - Data storage
  - Pre-training

	Dataset Info		Data-Stream Parameters			Pretrained
	Classes	Image-type	Tasks	Iterations	Batch size	Models?
MNIST	10	28x28, grey	5	2000	128	-
CIFAR-10	10	32x32, RGB	5	5000	256	-
CIFAR-100	100	32x32, RGB	10	5000	256	ConvLayers
CORe50	10	128x128, RGB	5	single pass	1	ResNet18

How to use a pre-trained model?

- Use the pre-trained model as a fixed feature extractor

- Train VAE models on the extracted features rather than on the raw inputs (i.e., with reconstruction loss in the feature space!)



- Reminiscent of recent studies performing generative replay in the feature space (van de Ven et al., 2020 *Nature Communications*; Liu et al., 2020 *CVPR-W*)

Results

Strategy	Method	MNIST	CIFAR-10	CIFAR-100	CORe50
Baselines	*None*	*19.92 (± 0.02)*	*18.74 (± 0.29)*	*7.96 (± 0.11)*	*18.65 (± 0.26)*
	Joint	*98.23 (± 0.04)*	*82.07 (± 0.15)*	*54.08 (± 0.27)*	*71.85 (± 0.30)*
Generative Replay	DGR	91.30 (± 0.60)	17.21 (± 1.88)	9.22 (± 0.24)	-
	BI-R	-	-	21.51 (± 0.25)	60.40 (± 1.04)
	BI-R + SI	-	-	34.38 (± 0.21)	62.68 (± 0.72)
Regularization	EWC	19.95 (± 0.05)	18.63 (± 0.29)	8.47 (± 0.09)	18.56 (± 0.31)
	SI	19.95 (± 0.11)	18.14 (± 0.36)	8.43 (± 0.08)	18.69 (± 0.26)
Bias-correction	CWR	32.48 (± 2.64)	18.37 (± 1.61)	21.90 (± 0.68)	40.28 (± 1.13)
	CWR+	37.20 (± 3.11)	22.32 (± 1.08)	9.34 (± 0.25)	40.12 (± 1.06)
	AR1	48.84 (± 2.55)	24.44 (± 1.08)	20.62 (± 0.45)	45.27 (± 1.02)
	Labels Trick	32.46 (± 1.95)	18.43 (± 1.31)	23.68 (± 0.26)	42.59 (± 1.03)
Other	SLDA	87.30 (± 0.02)	38.35 (± 0.03)	44.49 (± 0.00)	**70.80 (± 0.00)**
Generative Classifier		**93.79 (± 0.08)**	**56.03 (± 0.04)**	**49.55 (± 0.06)**	**70.81 (± 0.11)**

All experiments were performed 10 times with different random seeds, reported is the mean (± SEM) accuracy over these runs. Code to replicate: https://github.com/GMvandeVen/class-incremental-learning.

Results

Strategy	Method	MNIST	CIFAR-10	CIFAR-100	CORe50
Baselines	*None*	*19.92 (± 0.02)*	*18.74 (± 0.29)*	*7.96 (± 0.11)*	*18.65 (± 0.26)*
	Joint	*98.23 (± 0.04)*	*82.07 (± 0.15)*	*54.08 (± 0.27)*	*71.85 (± 0.30)*
Generative Replay	DGR	91.30 (± 0.60)	17.21 (± 1.88)	9.22 (± 0.24)	-
	BI-R	-	-	21.51 (± 0.25)	60.40 (± 1.04)
	BI-R + SI	-	-	34.38 (± 0.21)	62.68 (± 0.72)
Regularization	EWC	19.95 (± 0.05)	18.63 (± 0.29)	8.47 (± 0.09)	18.56 (± 0.31)
	SI	19.95 (± 0.11)	18.14 (± 0.36)	8.43 (± 0.08)	18.69 (± 0.26)
Bias-correction	CWR	32.48 (± 2.64)	18.37 (± 1.61)	21.90 (± 0.68)	40.28 (± 1.13)
	CWR+	37.20 (± 3.11)	22.32 (± 1.08)	9.34 (± 0.25)	40.12 (± 1.06)
	AR1	48.84 (± 2.55)	24.44 (± 1.08)	20.62 (± 0.45)	45.27 (± 1.02)
	Labels Trick	32.46 (± 1.95)	18.43 (± 1.31)	23.68 (± 0.26)	42.59 (± 1.03)
Other	SLDA	87.30 (± 0.02)	38.35 (± 0.03)	44.49 (± 0.00)	**70.80 (± 0.00)**
Generative Classifier		**93.79 (± 0.08)**	**56.03 (± 0.04)**	**49.55 (± 0.06)**	**70.81 (± 0.11)**

All experiments were performed 10 times with different random seeds, reported is the mean (± SEM) accuracy over these runs. Code to replicate: https://github.com/GMvandeVen/class-incremental-learning.

Generative classification *vs*. generative replay

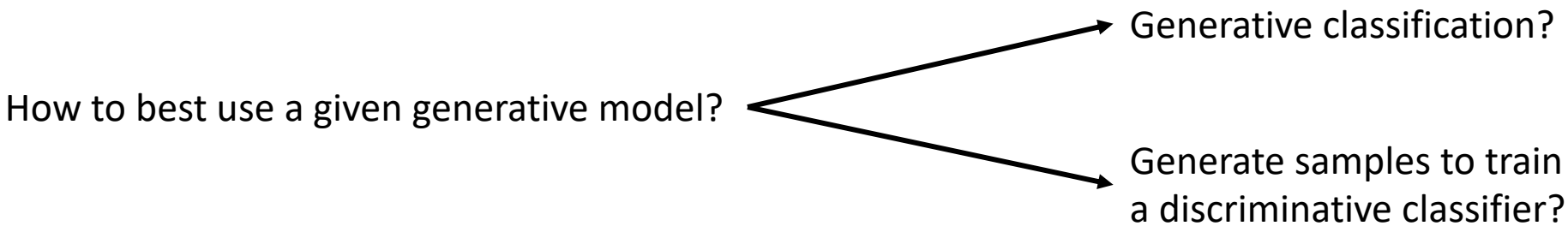
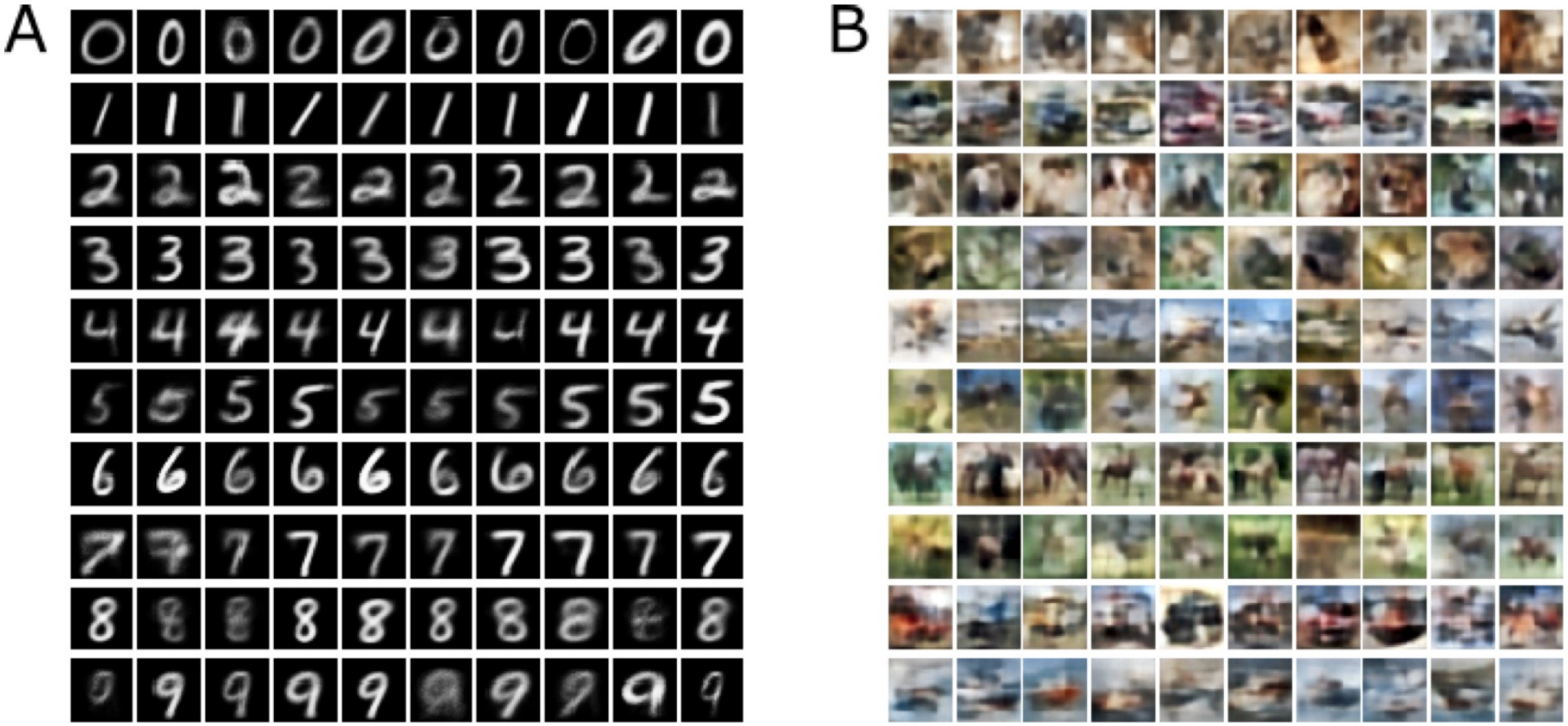


Table 3. Comparison of the performance of the generative classifier with the performance of a softmax-based classifier discriminatively trained on samples from the VAE models of the generative classifier. Shown is the test accuracy (as %) over all classes. All experiments were performed 10 times with different random seeds, reported are the means (± SEMs) over these runs.

	MNIST	**CIFAR-10**	**CIFAR-100**	**CORe50**
Generative classifier	**93.79 (± 0.08)**	**56.03 (± 0.04)**	**49.55 (± 0.06)**	**70.81 (± 0.11)**
Discriminative classifier trained on generated samples	85.93 (± 0.43)	13.71 (± 0.61)	33.84 (± 0.14)	47.86 (± 1.77)

Quality of the generative models underlying the generative classifier



Samples randomly drawn from the VAE models of the generative classifier for (A) MNIST and (B) CIFAR-10.

Results

Strategy	Method	MNIST	CIFAR-10	CIFAR-100	CORe50
Baselines	*None*	*19.92 (± 0.02)*	*18.74 (± 0.29)*	*7.96 (± 0.11)*	*18.65 (± 0.26)*
	Joint	*98.23 (± 0.04)*	*82.07 (± 0.15)*	*54.08 (± 0.27)*	*71.85 (± 0.30)*
Generative Replay	DGR	91.30 (± 0.60)	17.21 (± 1.88)	9.22 (± 0.24)	-
	BI-R	-	-	21.51 (± 0.25)	60.40 (± 1.04)
	BI-R + SI	-	-	34.38 (± 0.21)	62.68 (± 0.72)
Regularization	EWC	19.95 (± 0.05)	18.63 (± 0.29)	8.47 (± 0.09)	18.56 (± 0.31)
	SI	19.95 (± 0.11)	18.14 (± 0.36)	8.43 (± 0.08)	18.69 (± 0.26)
Bias-correction	CWR	32.48 (± 2.64)	18.37 (± 1.61)	21.90 (± 0.68)	40.28 (± 1.13)
	CWR+	37.20 (± 3.11)	22.32 (± 1.08)	9.34 (± 0.25)	40.12 (± 1.06)
	AR1	48.84 (± 2.55)	24.44 (± 1.08)	20.62 (± 0.45)	45.27 (± 1.02)
	Labels Trick	32.46 (± 1.95)	18.43 (± 1.31)	23.68 (± 0.26)	42.59 (± 1.03)
Other	SLDA	87.30 (± 0.02)	38.35 (± 0.03)	44.49 (± 0.00)	**70.80 (± 0.00)**
Generative Classifier		**93.79 (± 0.08)**	**56.03 (± 0.04)**	**49.55 (± 0.06)**	**70.81 (± 0.11)**

All experiments were performed 10 times with different random seeds, reported is the mean (± SEM) accuracy over these runs. Code to replicate: https://github.com/GMvandeVen/class-incremental-learning.

SLDA: a generative classifier in disguise

- SLDA (Hayes & Kanan, 2020 *CVPR-W*) performs incremental linear discriminant analysis to the features extracted by a fixed, pre-trained deep neural network
- For MNIST and CIFAR-10, we applied SLDA directly on the pixel space

- SLDA can be interpreted as a generative classifier
  - SLDA learns a mean vector $\boldsymbol{\mu}_y$ for each class y and a covariance matrix Σ that is shared between all classes, and the generative model that SLDA implicitly assumes for each class y is given by $p(\boldsymbol{x}|y) = N(\boldsymbol{x};\ \boldsymbol{\mu}_y, \Sigma)$

- SLDA can only learn a linear classifier
- SLDA's performance can be seen as the minimal attainable performance for a generative classifier, upon which can be improved when sufficient data is available

Results: a case for MNIST?

Strategy	Method	MNIST	CIFAR-10	CIFAR-100	CORe50
Baselines	*None*	*19.92 (± 0.02)*	*18.74 (± 0.29)*	*7.96 (± 0.11)*	*18.65 (± 0.26)*
	Joint	*98.23 (± 0.04)*	*82.07 (± 0.15)*	*54.08 (± 0.27)*	*71.85 (± 0.30)*
Generative Replay	DGR	91.30 (± 0.60)	17.21 (± 1.88)	9.22 (± 0.24)	-
	BI-R	-	-	21.51 (± 0.25)	60.40 (± 1.04)
	BI-R + SI	-	-	34.38 (± 0.21)	62.68 (± 0.72)
Regularization	EWC	19.95 (± 0.05)	18.63 (± 0.29)	8.47 (± 0.09)	18.56 (± 0.31)
	SI	19.95 (± 0.11)	18.14 (± 0.36)	8.43 (± 0.08)	18.69 (± 0.26)
Bias-correction	CWR	32.48 (± 2.64)	18.37 (± 1.61)	21.90 (± 0.68)	40.28 (± 1.13)
	CWR+	37.20 (± 3.11)	22.32 (± 1.08)	9.34 (± 0.25)	40.12 (± 1.06)
	AR1	48.84 (± 2.55)	24.44 (± 1.08)	20.62 (± 0.45)	45.27 (± 1.02)
	Labels Trick	32.46 (± 1.95)	18.43 (± 1.31)	23.68 (± 0.26)	42.59 (± 1.03)
Other	SLDA	87.30 (± 0.02)	38.35 (± 0.03)	44.49 (± 0.00)	**70.80 (± 0.00)**
Generative Classifier		**93.79 (± 0.08)**	**56.03 (± 0.04)**	**49.55 (± 0.06)**	**70.81 (± 0.11)**

All experiments were performed 10 times with different random seeds, reported is the mean (± SEM) accuracy over these runs. Code to replicate: https://github.com/GMvandeVen/class-incremental-learning.

Limitations / discussion / future work

- Inference with generative classifiers is slow, as likelihood must be computed/estimated for each possible class
  → Likelihood estimation can be (a lot) more efficient
  → Classification decisions could be made hierarchical

- How scalable is learning a new generative model for each class?
  → In the comparison we controlled for *total* number of parameters
  → Share parts of the generative models (e.g., using existing techniques for task-incremental learning)

One-sentence summary

- Generative classification is a promising, “rehearsal-free” strategy for class-incremental learning

Funding acknowledgements

This research project has been supported by the Lifelong Learning Machines (L2M) program of the Defence Advanced Research Projects Agency (DARPA) via contract number HR0011-18-2-0025 and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. Disclaimer: The views and conclusions contained in this presentation are those of the presenter and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, IARPA, DoI/IBC, or the U.S. Government.

References

- Hayes TL, Kanan C (2020) Lifelong machine learning with deep streaming linear discriminant analysis. *CVPR workshop*: 220–221.
- Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R (2017) Overcoming catastrophic forgetting in neural networks. *PNAS* **114**: 3521-3526.
- Liu X, Wu C, Menta M, Herranz L, Raducanu B, Bagdanov AD, Jui S, van de Weijer J (2020) Generative feature replay for class-incremental learning. *CVPR workshop*: 226–227.
- Lomonaco V, Maltoni D (2017) Core50: a new dataset and benchmark for continuous object recognition. *CoRL*: 17–26.
- Maltoni D, Lomonaco V (2019) Continuous learning in single-incremental-task scenarios. *Neural Networks*, **116**: 56–73.
- Shin H, Lee JK, Kim J, Kim J (2017) Continual learning with deep generative replay. *NeurIPS*: 2994-3003.
- van de Ven GM, Siegelmann HT, Tolias AS (2020) Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications* **11**: 4069.
- van de Ven GM, Tolias AS (2019) Three scenarios for continual learning. *ArXiv preprint*: 1904.07734.
- Zenke F, Poole B, Ganguli S (2017) Continual learning through synaptic intelligence. *ICML*: 3987-3995.
- Zeno C, Golan I, Hoffer E, Soudry D (2019) Task agnostic continual learning using online variational bayes. *ArXiv preprint*: 1803.10123v3.