SABE: Continual Learning

Scenarios ?
Assumptions ??
Benchmarks ???
Evaluation ????

Sanket Vaibhav Mehta (SVM)

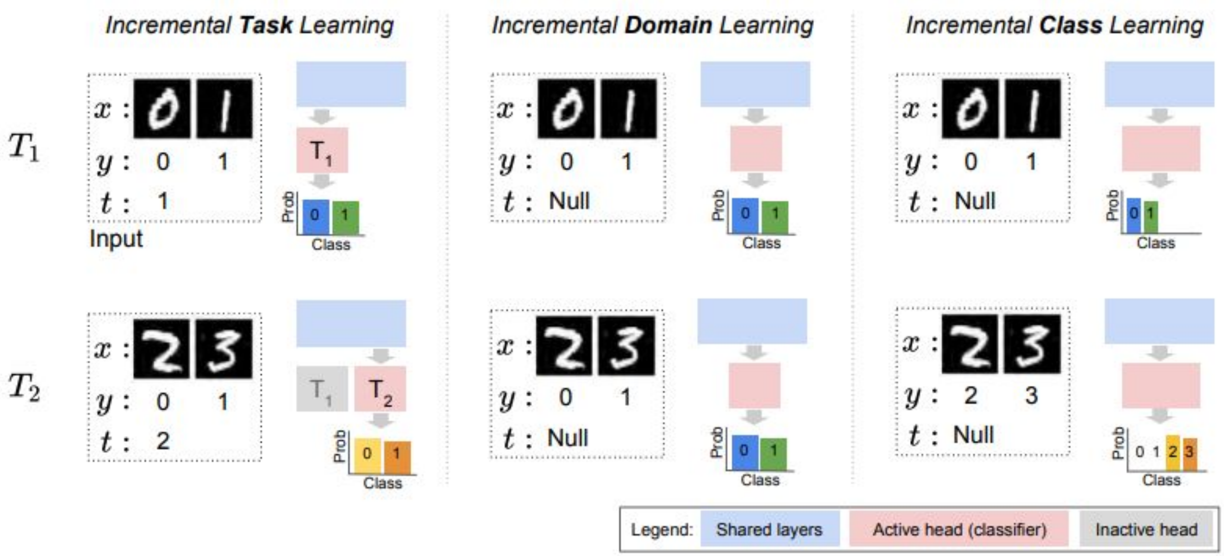
Scenarios

Table 1: The continual learning scenarios categorized by the difference between the old and new task. $P(X)$: The distribution of input data. $P(Y)$: The distribution of target labels. $\{Y_1\} \neq \{Y_2\}$: The labels are from a disjoint space which is differentiated by task identity. S: Single-headed model. M: Multi-headed model. I: Known task identity.

Learning scenario	Old Task (T_1) versus New Task (T_2)			Remark
	$P(X_1) \neq P(X_2)$	$P(Y_1) \neq P(Y_2)$	$\{Y_1\} \neq \{Y_2\}$	
Non-incremental learning				
Incremental domain learning	✓			S
Incremental class learning	✓	✓		S
Incremental task learning	✓	✓	✓	M, I

[1] Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines, Continual Learning Workshop, NeurIPS 2018

Scenarios



[1] Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines, Continual Learning Workshop, NeurIPS 2018

Assumptions

1. Cross-task resemblances (confident incorrect predictions)
2. Shared output head i.e., single vs. multi-head (difficulty of the problem under consideration)
3. (No) Test-time assumed task labels (Is it continual learning anymore ??)
4. (No) Retraining on old tasks (Does memory violate privacy laws?)
5. More than two-tasks (Two-task transfer doesn't guarantee success)
6. No. of training examples (Humans learn from few examples per task)
7. Revisiting dataset multiple times (Humans observe examples only once)

[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018
[3] Gradient Episodic Memory for Continual Learning, NIPS 2017

Challenging Scenarios

1. Unclear task demarcation (currently task boundaries are assumed to be known)
2. Continuous tasks (currently tasks are discrete different)
3. Overlapping tasks (current task output spaces have minimal overlap)
4. Long task sequences
5. Time constraints (faster adaptation to new tasks is desirable)
6. Memory constraints (small fixed memory is desirable)

[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018
[3] Gradient Episodic Memory for Continual Learning, NIPS 2017

Aspects of CL

1. **Catastrophic interference or Catastrophic forgetting**

   "Continual learning is at its hardest when new tasks resemble old ones, enough that the model makes confident but incorrect predictions on the new data." [2]

2. **Strong forward (or backward) transfer**

   "when the tasks in the continuum are related, there exists an opportunity for transfer learning. This would translate into faster learning of new tasks, as well as performance improvements in old tasks" [3]

[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018
[3] Gradient Episodic Memory for Continual Learning, NIPS 2017

CL Approaches

1. **Prior-focused** (use regularization to create “elastic parameters”)
    a. Elastic Weight Consolidation (EWC)
    b. Synaptic Intelligence (SI)
    c. Variational Continual Learning (VCL)
2. **Memory Replay** (or Pseudo-rehearsals)
    a. Gradient Episodic Memory (GEM/ Average-GEM)
    b. VCL + Coreset
    c. Deep Generative Replay (DGR)
3. **Dynamic architectures**
    a. Progressive Networks

[4] Continual Lifelong Learning with Neural Networks: A Review, Neural Networks, 2019

Benchmark datasets

1. Permuted MNIST
2. Rotate MNIST
3. Split MNIST
4. Fashion MNIST
5. SVHN

6. CUB-200
7. AudioSet-100
8. CIFAR-100
9. IMAGENET-100

10. Core50

	MNIST	CUB-200	AudioSet
Classification Task	Gray Image	RGB Image	Audio
Classes	10	200	100
Feature Shape	784	2,048	1,280
Train Samples	50,000	5,994	28,779
Test Samples	10,000	5,794	5,523
Train Samples/Class	5,421-6,742	29-30	250-300
Test Samples/Class	892-1,135	11-30	43-62

Table 1: Dataset Specifications

[5] Measuring Catastrophic Forgetting in Neural Networks, AAAI, 2018

Evaluation Metrics

1. Average Accuracy (ACC)
2. Forward Transfer (FWT) - the influence of task ‘t’ on a future task
3. Backward Transfer (BWT) - the influence of task ‘t’ on a previous task

Shortcomings of Permuted MNIST for CL

- Dataset for each task is constructed from the MNIST data but with the pixels of each digit randomly permuted (same permutation applied to all images for a given task)
- Permuted MNIST is always performed single-headed (10-way classification)
- This benchmark masks weak-points of the current approaches

	ENTROPY
Split	0.003
Permuted	0.453

Figure 2: Average entropy of predictions on Task B, early in training; Note the 2 orders of magnitude difference between the two settings. Entropy is much higher in the Permuted setting.

[6] An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks, 2013
[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018

Shortcomings of Permuted MNIST for CL

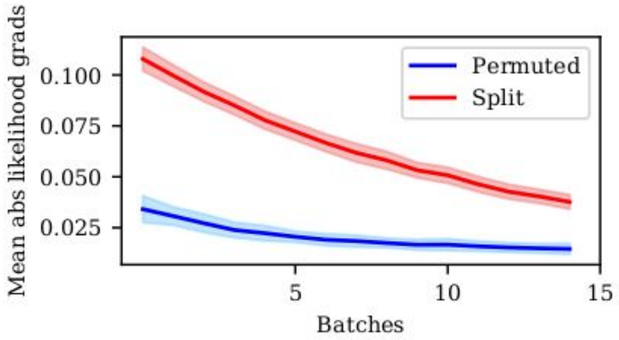


Figure 3: Early in training Task B, the likelihood term of gradients on the final layer is unusually low in the Permuted setting because permuted digits do not resemble any digits from Task A. This makes continual learning unrealistically easy in this evaluation. Averaged over 100 runs, shading is one standard deviation.

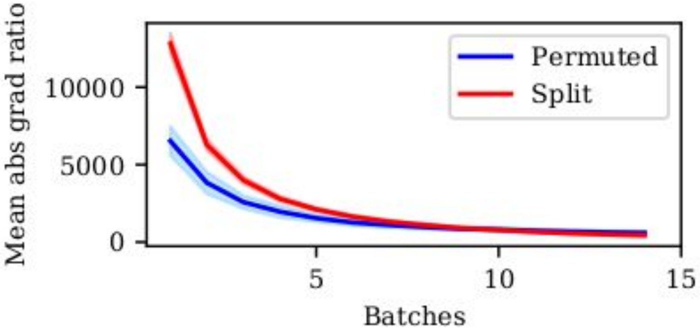


Figure 4: Early in training Task B, the ratio of the likelihood term of gradients the final layer against the prior term is much lower in the Permuted setting. This reduces forgetting because the prior has a larger impact on learning. Averaged over 100 runs, shading is one standard deviation.

[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018

Split MNIST and Fashion MNIST

Task Incremental Learning (Task identifiers are provided⇒**Very Easy Evaluation**)

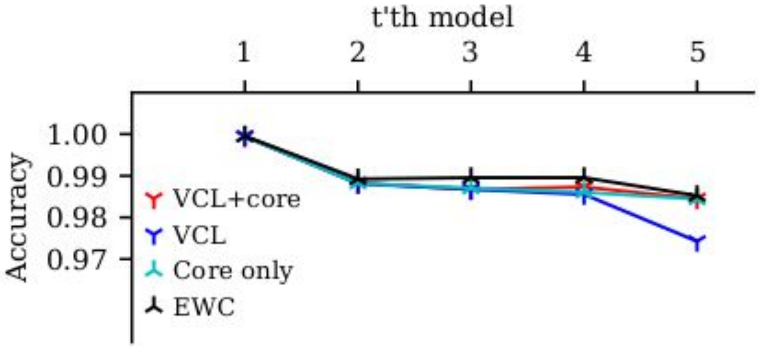


Figure 5: **Multi-headed Split MNIST.** VCL and EWC appear effective. VCL works with or without coresets, and coresets work well on their own.

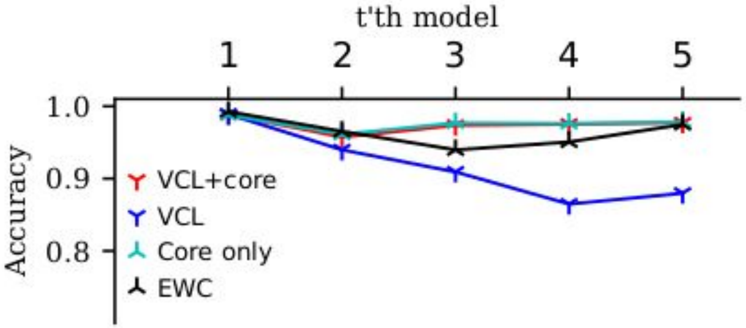


Figure 7: **Multi-headed Split Fashion MNIST.** As with MNIST, all algorithms perform similarly, though VCL performs very slightly worse without coresets.

[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018

Split MNIST and Fashion MNIST

Class Incremental Learning (**Prior-based approaches fail terribly**)

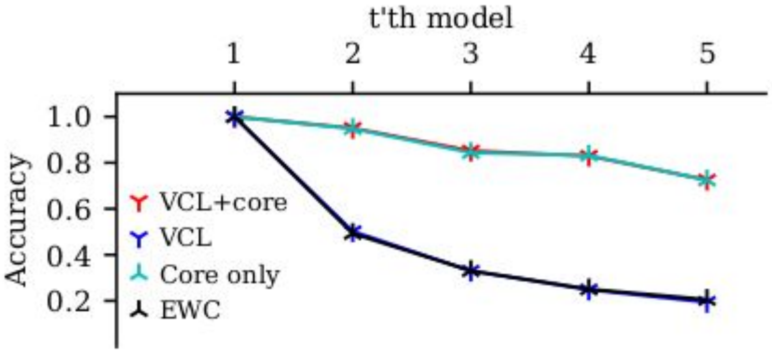


Figure 6: **Single-headed Split MNIST.** Coresets seem to cause the hybrid model's good performance: coresets alone perform the same. VCL and EWC forget old tasks completely. The single-headed setting reveals blind-spots from the multi-headed setting.

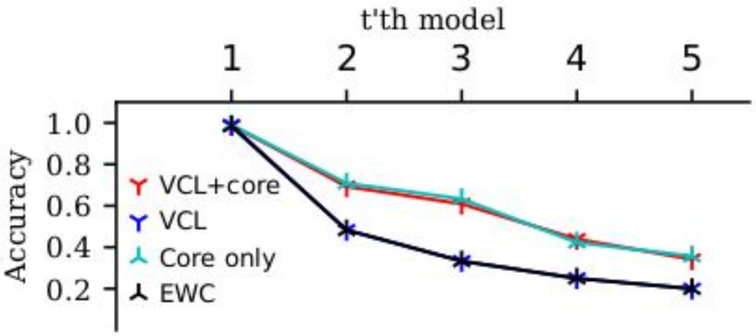
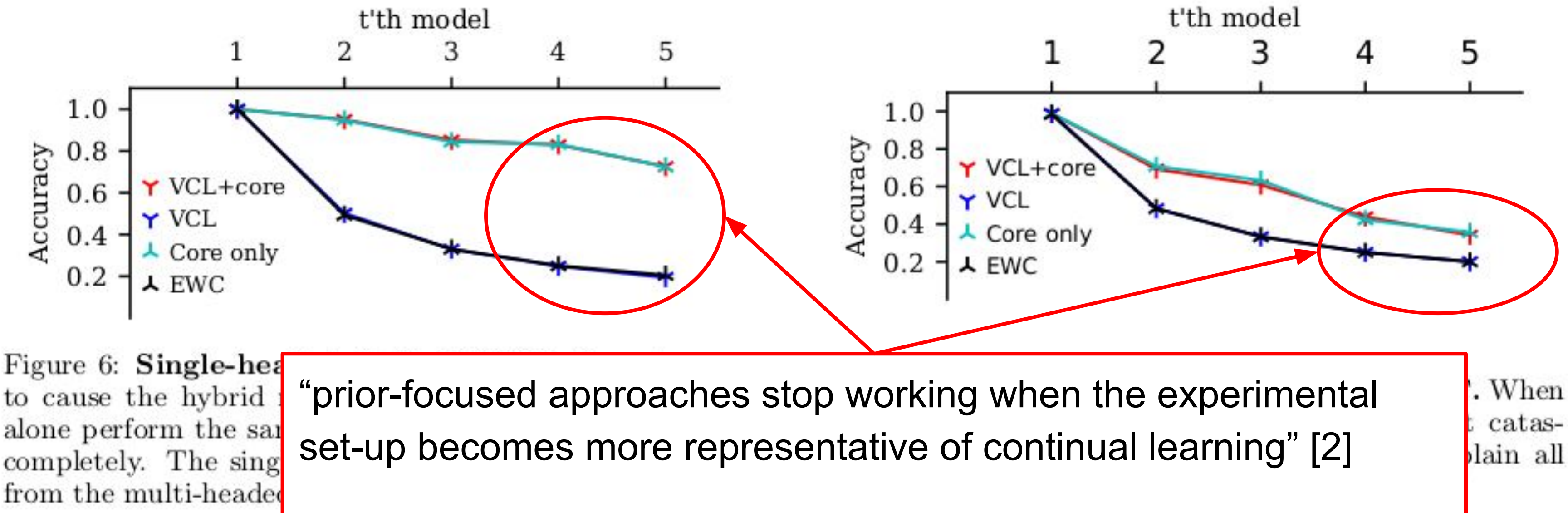


Figure 8: **Single-headed Split Fashion MNIST.** When single-headed, VCL and EWC no longer prevent catastrophic forgetting. Coresets do, and seem to explain all of VCL with coreset's performance.

[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018

Split MNIST and Fashion MNIST

Class Incremental Learning (**Prior-based approaches fail terribly**)

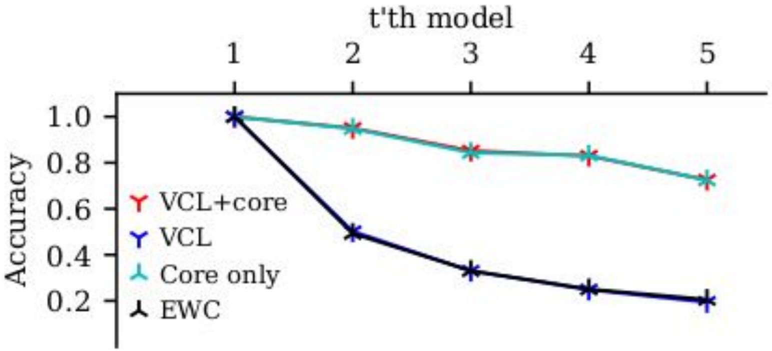


"prior-focused approaches stop working when the experimental set-up becomes more representative of continual learning" [2]

[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018

Split MNIST and Fashion MNIST

Class Incremental Learning (**Prior-based approaches fail terribly**)



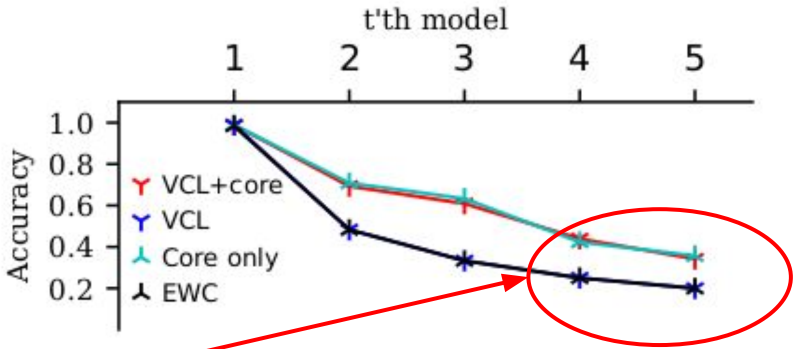


Figure 6: **Single-hea**
to cause the hybrid
alone perform the sa
completely. The sing
from the multi-heade

. When
catas-
plain all

Fashion MNIST is relatively harder benchmark in case of Class Incremental Learning scenario

[2] Towards Robust Evaluations of Continual Learning, LLARLA Workshop, ICML 2018

Split MNIST

Table 2: The average accuracy (%, higher is better) of all seen tasks after learning the task sequence generated by Split MNIST (Figure 1). The *Memory* column means whether a method uses a memory mechanism, which further divides the methods into two groups in the comparison. The total static memory overhead is controlled to be the same among L2, Naive rehearsal, Naive rehearsal-C, online EWC, SI, MAS, GEM, and DGR. Each value is the average of 10 runs.

	Method	Memory	Incremental task learning	Incremental domain learning	Incremental class learning
Baselines	Adam		93.46 ± 2.01	55.16 ± 1.38	19.71 ± 0.08
	SGD		97.98 ± 0.09	63.20 ± 0.35	19.46 ± 0.04
	Adagrad		98.06 ± 0.53	58.08 ± 1.06	19.82 ± 0.09
	L2		98.18 ± 0.96	66.00 ± 3.73	22.52 ± 1.08
	Naive rehearsal	✓	99.40 ± 0.08	95.16 ± 0.49	90.78 ± 0.85
	Naive rehearsal-C	✓	$\mathbf{99.57} \pm 0.07$	$\mathbf{97.11} \pm 0.34$	$\mathbf{95.59} \pm 0.49$
Continual learning methods	EWC		97.70 ± 0.81	58.85 ± 2.59	19.80 ± 0.05
	Online EWC		98.04 ± 1.10	57.33 ± 1.44	19.77 ± 0.04
	SI		98.56 ± 0.49	64.76 ± 3.09	19.67 ± 0.09
	MAS		99.22 ± 0.21	68.57 ± 6.85	19.52 ± 0.29
	LwF		99.60 ± 0.03	71.02 ± 1.26	24.17 ± 0.33
	GEM	✓	98.42 ± 0.10	96.16 ± 0.35	92.20 ± 0.12
	DGR	✓	99.47 ± 0.03	95.74 ± 0.23	91.24 ± 0.33
	RtF	✓	$\mathbf{99.66} \pm 0.03$	$\mathbf{97.31} \pm 0.11$	$\mathbf{92.56} \pm 0.21$
Offline (upper bound)			99.52 ± 0.16	98.59 ± 0.15	97.53 ± 0.30

[2] Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines, Continual Learning Workshop, NeurIPS 2018

Split MNIST

Prior-focused approaches (EWC, SI) results are similar to VCL results

Table 2: The average accuracy (%, higher is better) of all seen tasks after learning the task sequence generated by Split MNIST (Figure 1). The *Memory* column means whether a method uses a memory mechanism, which further divides the methods into two groups in the comparison. The total static memory overhead is controlled to be the same among L2, Naive rehearsal, Naive rehearsal-C, online EWC, SI, MAS, GEM, and DGR. Each value is the average of 10 runs.

	Method	Memory	Incremental task learning	Incremental domain learning	Incremental class learning
Baselines	Adam		93.46 ± 2.01	55.16 ± 1.38	19.71 ± 0.08
	SGD		97.98 ± 0.09	63.20 ± 0.35	19.46 ± 0.04
	Adagrad		98.06 ± 0.53	58.08 ± 1.06	19.82 ± 0.09
	L2		98.18 ± 0.96	66.00 ± 3.73	22.52 ± 1.08
	Naive rehearsal	✓	99.40 ± 0.08	95.16 ± 0.49	90.78 ± 0.85
	Naive rehearsal-C	✓	$\mathbf{99.57} \pm 0.07$	$\mathbf{97.11} \pm 0.34$	$\mathbf{95.59} \pm 0.49$
Continual learning methods	EWC		97.70 ± 0.81	58.85 ± 2.59	19.80 ± 0.05
	Online EWC		98.04 ± 1.10	57.33 ± 1.44	19.77 ± 0.04
	SI		98.56 ± 0.49	64.76 ± 3.09	19.67 ± 0.09
	MAS		99.22 ± 0.21	68.57 ± 6.85	19.52 ± 0.29
	LwF		99.60 ± 0.03	71.02 ± 1.26	24.17 ± 0.33
	GEM	✓	98.42 ± 0.10	96.16 ± 0.35	92.20 ± 0.12
	DGR	✓	99.47 ± 0.03	95.74 ± 0.23	91.24 ± 0.33
	RtF	✓	$\mathbf{99.66} \pm 0.03$	$\mathbf{97.31} \pm 0.11$	$\mathbf{92.56} \pm 0.21$
Offline (upper bound)			99.52 ± 0.16	98.59 ± 0.15	97.53 ± 0.30

[2] Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines, Continual Learning Workshop, NeurIPS 2018

Split MNIST

MNIST seems too easy benchmark for Incremental Task Learning Scenario

Table 2: The average accuracy (%, higher is better) of all seen tasks after learning the task sequence generated by Split MNIST (Figure 1). The *Memory* column means whether a method uses a memory mechanism, which further divides the methods into two groups in the comparison. The total static memory overhead is controlled to be the same among L2, Naive rehearsal, Naive rehearsal-C, online EWC, SI, MAS, GEM, and DGR. Each value is the average of 10 runs.

	Method	Memory	Incremental task learning	Incremental domain learning	Incremental class learning
Baselines	Adam		93.46 ± 2.01	55.16 ± 1.38	19.71 ± 0.08
	SGD		97.98 ± 0.09	63.20 ± 0.35	19.46 ± 0.04
	Adagrad		98.06 ± 0.53	58.08 ± 1.06	19.82 ± 0.09
	L2		98.18 ± 0.96	66.00 ± 3.73	22.52 ± 1.08
	Naive rehearsal	✓	99.40 ± 0.08	95.16 ± 0.49	90.78 ± 0.85
	Naive rehearsal-C	✓	**99.57** ± 0.07	**97.11** ± 0.34	**95.59** ± 0.49
Continual learning methods	EWC		97.70 ± 0.81	58.85 ± 2.59	19.80 ± 0.05
	Online EWC		98.04 ± 1.10	57.33 ± 1.44	19.77 ± 0.04
	SI		98.56 ± 0.49	64.76 ± 3.09	19.67 ± 0.09
	MAS		99.22 ± 0.21	68.57 ± 6.85	19.52 ± 0.29
	LwF		99.60 ± 0.03	71.02 ± 1.26	24.17 ± 0.33
	GEM	✓	98.42 ± 0.10	96.16 ± 0.35	92.20 ± 0.12
	DGR	✓	99.47 ± 0.03	95.74 ± 0.23	91.24 ± 0.33
	RtF	✓	**99.66** ± 0.03	**97.31** ± 0.11	**92.56** ± 0.21
Offline (upper bound)			99.52 ± 0.16	98.59 ± 0.15	97.53 ± 0.30

[2] Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines, Continual Learning Workshop, NeurIPS 2018

Split MNIST

Table 2: The average accuracy (%, higher is better) of all seen tasks after learning the task sequence generated by Split MNIST (Figure 1). The *Memory* column means whether a method uses a memory mechanism, which further divides the methods into two groups in the comparison. The total static memory overhead is controlled to be the same among L2, Naive rehearsal, Naive rehearsal-C, online EWC, SI, MAS, GEM, and DGR. Each value is the average of 10 runs.

	Method	Memory	Incremental task learning	Incremental domain learning	Incremental class learning
Baselines	Adam		93.46 ± 2.01	55.16 ± 1.38	19.71 ± 0.08
	SGD		97.98 ± 0.09	63.20 ± 0.35	19.46 ± 0.04
	Adagrad		98.06 ± 0.53	58.08 ± 1.06	19.82 ± 0.09
	L2		98.18 ± 0.96	66.00 ± 3.73	22.52 ± 1.08
	Naive rehearsal	✓	99.40 ± 0.08	95.16 ± 0.49	90.78 ± 0.85
	Naive rehearsal-C	✓	$\mathbf{99.57} \pm 0.07$	$\mathbf{97.11} \pm 0.34$	$\mathbf{95.59} \pm 0.49$
Continual learning methods	EWC		97.70 ± 0.81	58.85 ± 2.59	19.80 ± 0.05
	Online EWC		98.04 ± 1.10	57.33 ± 1.44	19.77 ± 0.04
	SI		98.56 ± 0.49	64.76 ± 3.09	19.67 ± 0.09
	MAS		99.22 ± 0.21	68.57 ± 6.85	19.52 ± 0.29
	LwF		99.60 ± 0.03	71.02 ± 1.26	24.17 ± 0.33
	GEM	✓	98.42 ± 0.10	96.16 ± 0.35	92.20 ± 0.12
	DGR	✓	99.47 ± 0.03	95.74 ± 0.23	91.24 ± 0.33
	RtF	✓	$\mathbf{99.66} \pm 0.03$	$\mathbf{97.31} \pm 0.11$	$\mathbf{92.56} \pm 0.21$
Offline (upper bound)			99.52 ± 0.16	98.59 ± 0.15	97.53 ± 0.30

This work [2] sets very high replay buffer size(s): 1.1k and 4.4k

[2] Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines, Continual Learning Workshop, NeurIPS 2018

Split MNIST

Table 2: The average accuracy (%, higher is better) of all seen tasks after learning the task sequence generated by Split MNIST (Figure 1). The *Memory* column means whether a method uses a memory mechanism, which further divides the methods into two groups in the comparison. The total static memory overhead is controlled to be the same among L2, Naive rehearsal, Naive rehearsal-C, online EWC, SI, MAS, GEM, and DGR. Each value is the average of 10 runs.

	Method	Memory	Incremental task learning	Incremental domain learning	Incremental class learning
Baselines	Adam		93.46 ± 2.01	55.16 ± 1.38	19.71 ± 0.08
	SGD		97.98 ± 0.09	63.20 ± 0.35	19.46 ± 0.04
	Adagrad		98.06 ± 0.53	58.08 ± 1.06	19.82 ± 0.09
	L2		98.18 ± 0.96	66.00 ± 3.73	22.52 ± 1.08
	Naive rehearsal	✓	99.40 ± 0.08	95.16 ± 0.49	90.78 ± 0.85
	Naive rehearsal-C	✓	$\mathbf{99.57} \pm 0.07$	$\mathbf{97.11} \pm 0.34$	$\mathbf{95.59} \pm 0.49$
Continual learning methods	EWC		97.70 ± 0.81	58.85 ± 2.59	19.80 ± 0.05
	Online EWC		98.04 ± 1.10	57.33 ± 1.44	19.77 ± 0.04
	SI		98.56 ± 0.49	64.76 ± 3.09	19.67 ± 0.09
	MAS		99.22 ± 0.21	68.57 ± 6.85	19.52 ± 0.29
	LwF		99.60 ± 0.03	71.02 ± 1.26	24.17 ± 0.33
	GEM	✓	98.42 ± 0.10	96.16 ± 0.35	92.20 ± 0.12
	DGR	✓	99.47 ± 0.03	95.74 ± 0.23	91.24 ± 0.33
	RtF	✓	$\mathbf{99.66} \pm 0.03$	$\mathbf{97.31} \pm 0.11$	$\mathbf{92.56} \pm 0.21$
Offline (upper bound)			99.52 ± 0.16	98.59 ± 0.15	97.53 ± 0.30

Generative Replay methods are only evaluated on simple domains like MNIST

[2] Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines, Continual Learning Workshop, NeurIPS 2018

Discussion

1. Prior-based approaches doesn’t prevent catastrophic forgetting (or datasets on which they are evaluated don’t render realistic continual learning scenarios)
2. MNIST Benchmarks are simpler when provided with task information (or multi-headed version).
3. MNIST Benchmarks are still challenging for Class and Domain Incremental Learning, especially Fashion MNIST
4. Once can also explore more realistic Domain Incremental Learning i.e., MNIST, USPS, Street View Digits
5. For Task Incremental Learning one should consider complex sequence of tasks
6. For Generative Replay methods one should consider evaluating on complex domains in terms of generative models

Questions?