Knowledge Capture and Replay for Continual Learning - Supplementary

Saisubramaniam Gopalakrishnan [1,4], Pranshu Ranjan Singh [1,4], Haytham Fayek [3],
Savitha Ramasamy [1,2,4], ArulMurugan Ambikapathi [1,2,4]
[1] Institute for Infocomm Research (I^{2}R), A*STAR, Singapore *
[2] Artificial Intelligence, Analytics And Informatics (AI^3), A*STAR, Singapore
[3] RMIT University, Australia
[4] CNRS@CREATE LTD, 1 Create Way, 08-01 CREATE Tower, Singapore 138602 †
{g_saisubramaniam, pranshurs, ramasamysa, arul}@i2r.a-star.edu.sg, haytham.fayek@ieee.org

A. Benchmarking Number of Flashcards

Increase in flashcards number shows improvement - examples for MNIST and Cifar10, as observed in Figure 1. Trend followed by flashcards is similar to the improvement shown with increasing coreset/exemplars.

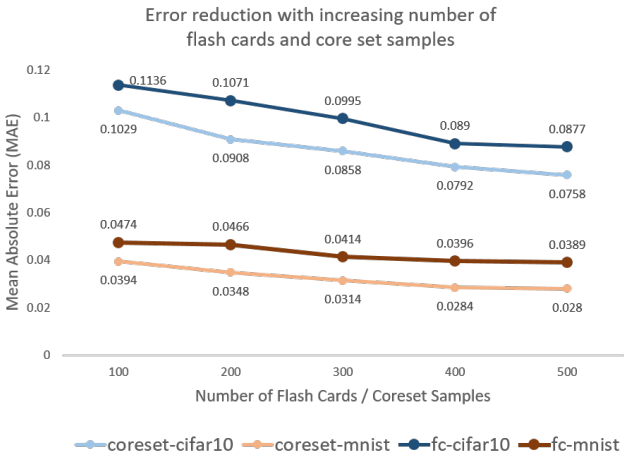


Figure 1. Increase in no. of flashcards / coreset improves performance - Quantitative error reduction in (i) Cifar10, (ii) MNIST.

B. Visualization of different starting initializations for flashcards construction

Extended visualization of each initialization - Maze patterns, Gaussian noise, and new task data as initialization, and the corresponding flashcards constructed after $1, 2, 3...8, 9, 10$ iterations are provided in this section. We can observe the diversity from flashcards originating from maze patterns, in comparison to the other two approaches.

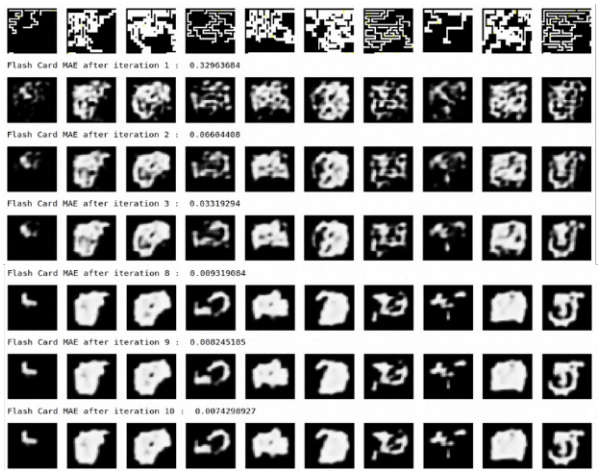


Figure 2. Different iterations of flashcards construction from maze patterns, after training autoencoder on MNIST

C. Evaluating Flashcards Derived Autoencoder - Are the Reconstructions good enough to be used for Classification?

Dataset	Original	Classifier 1	Classifier 2
MNIST	0.9940	0.9921	0.9869
Fashion MNIST	0.9215	0.9126	0.8978
Cifar10	0.8003	0.6295	0.5659

Table 1. Building a classifier using the reconstructions from flashcard trained network. Reported accuracy averaged over 5 runs. Cifar10 accuracy is lower due to relatively higher AE recon error.

Let AE1 be trained on original images D_{T_1} and AE-Flash1 be trained using flashcards D_{f_1} from AE1. Let the respective reconstructions (after training) be $\widehat{D_{T_1}}$ and $\widehat{D_{F_1}}$.

*This research was partially funded through SC20/19-128310-CORE.
†This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

Figure 3. Different iterations of flashcards construction from maze patterns, after training autoencoder on Fashion MNIST

Figure 4. Different iterations of flashcards construction from maze patterns, after training autoencoder on Cifar10

Figure 5. Different iterations of flashcards construction from maze patterns, after training autoencoder on SVHN

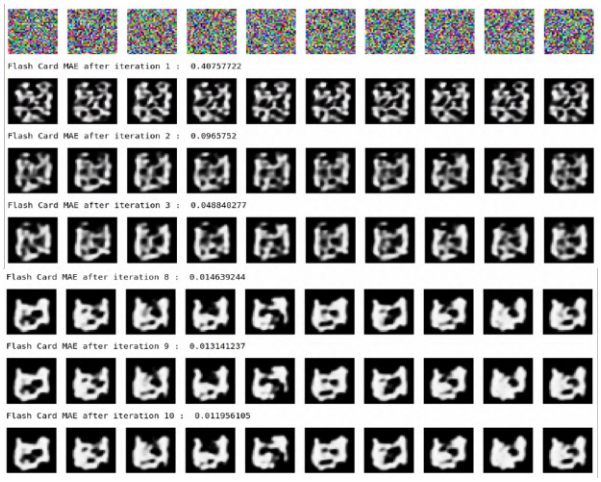


Figure 6. Different iterations of flashcards construction from Gaussian noise, after training autoencoder on MNIST

Next, train two classifiers (VGG16), Classifier1 and Classifier2 using $\widehat{D_{T_1}}$ and $\widehat{D_{F_1}}$, respectively, and compare their performance on independent test set of original images. The results tabulated in Table 1 shows that flashcards trained autoencoder networks are capable of providing reconstructions, that perform reasonably well as inputs for classification.

D. AutoEncoder Architecture Selection

We train several AE architectures on Cifar10 dataset to compare the performance of flashcards for reconstruction. Table 2 provides the details about various model architectures and the corresponding test Mean Absolute Error (MAE) on training using the original dataset (Original MAE) and using Flashcards generated from the trained AE (Flashcards MAE), respectively.

We choose the architecture *Blk_4_fil_64* for all our experiments for reconstruction and denoising tasks. *Blk_4_fil_64* architecture obtains 0.0512 Original MAE and 0.0570 Flashcards MAE. *Blk_3_fil_64* and *Blk_2_fil_32* achieve better Original MAE/Flashcards MAE than *Blk_4_fil_64*, but both these architectures use higher latent space size (1024 and 2048). The improvement in reconstruction MAE may be attributed to such high latent space dimensions which might not encode useful information and just act as a copy function. Hence, we choose the *Blk_4_fil_64* architecture.

E. Benchmarking Flashcards based on Iterations Processed

Table 3 provides the benchmark errors on using 50,000 flashcards obtained via different iterations, when used for training a network from scratch.

Arch. Type	Model Params.	Latent Space	Num. Blocks	Num. Filters	Original MAE	Flashcards MAE
Blk_4_fil_16	24,083	64 (48x reduction)	4	16	0.0787 ± 0.0002	0.0963 ± 0.0004
Blk_4_fil_32	94,243	128 (24x reduction)	4	32	0.0640 ± 0.0002	0.0725 ± 0.0009
Blk_4_fil_64	372,803	**256 (12x reduction)**	4	64	0.0512 ± 0.0004	0.0570 ± 0.0006
Blk_4_fil_128	1,482,883	512 (6x reduction)	4	128	0.2062 ± 0.0000	0.2445 ± 0.0476
Blk_3_fil_64	298,947	1024 (3x reduction)	3	64	0.0437 ± 0.0003	0.0599 ± 0.0067
Blk_2_fil_32	57,251	2048 (1.5x reduction)	2	32	0.0358 ± 0.0008	0.0389 ± 0.0015

Table 2. Architecture selection for AutoEncoder (AE). We train several AE architectures on Cifar10 dataset in order to compare the performance of flashcards for reconstruction. Various details about the architecture such as Model Params. (Number of trainable weights and biases), Latent Space (Size of latent space/bottleneck layer and its reduction rate versus image space), Num. Blocks (Number of convolution + pooling blocks in Encoder), and Num. Filters (Number of filters in convolution layers) are also provided. Original MAE is the Cifar10 test MAE on AE trained using Cifar10 train dataset. Flashcards MAE is the Cifar10 test MAE on AE trained using the flashcards obtained from given AE. The reported standard deviation for the scores are obtained over 5 experimental runs.

Dataset	Original	Flashcard 1 iteration	Flashcard 3 iterations	Flashcard 5 iterations	Flashcard 10 iterations	Flashcard 12 iterations	Flashcard 15 iterations
Cifar10	0.0564 $\pm$ 0.0014	0.1300 $\pm$ 0.0156	0.0713 $\pm$ 0.0010	0.0708 $\pm$ 0.0027	0.0686 $\pm$ 0.0042	0.0732 $\pm$ 0.0109	0.0740 $\pm$ 0.0097
MNIST	0.0184 $\pm$ 0.0014	0.0880 $\pm$ 0.0055	0.0696 $\pm$ 0.0018	0.0675 $\pm$ 0.0007	0.0491 $\pm$ 0.0053	0.0419 $\pm$ 0.0033	0.0417 $\pm$ 0.0027
Fashion MNIST	0.0259 $\pm$ 0.0003	0.1194 $\pm$ 0.0010	0.0451 $\pm$ 0.0003	0.0437 $\pm$ 0.0003	0.0440 $\pm$ 0.0006	0.0435 $\pm$ 0.0007	0.0465 $\pm$ 0.0019

Table 3. Benchmarking effect of iterations for reconstruction of different datasets, using 50k flashcards.

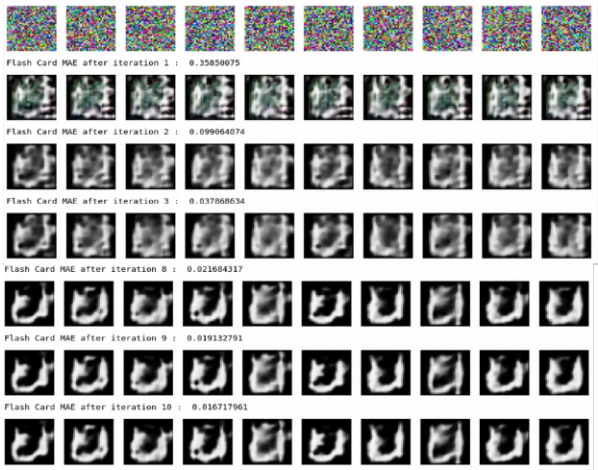


Figure 7. Different iterations of flashcards construction from Gaussian noise, after training autoencoder on Fashion MNIST

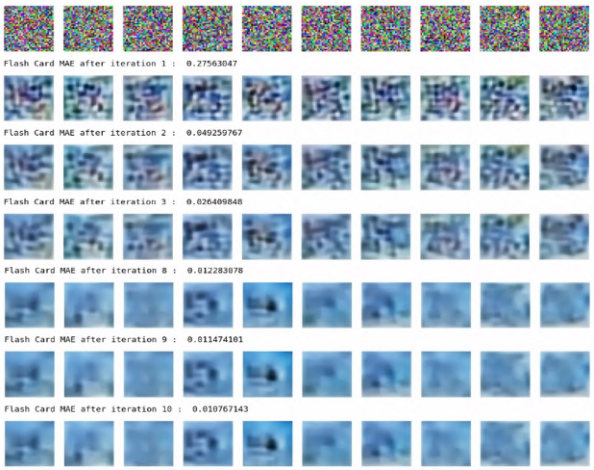


Figure 8. Different iterations of flashcards construction from Gaussian noise, after training autoencoder on Cifar10

F. Sequence5 Continual Reconstruction

We provide visuals for each task in Sequence5 showing how each method handles forgetting. Figure 14 shows Sequential Fine Tuning (SFT) is the naive approach and suffers the most. It can be observed that reconstructions are empty, this is because of the network parameters at the start of task 5, which prevents it to learn the current Omniglot task itself. Figure 15 shows the effect of replay with 500 real samples (coreset). 500 samples were chosen as their memory matches the AE network parameters of 1.5MB. From the experimental results, it is observed that 500 samples are not sufficient to beat flashcards. Figure 19 has individual graphs for different methods show the variation of test Mean Absolute Error (MAE) on current task dataset after observing the data for sequence of tasks.

Figure 16 is based on VAE trained in CL fashion, maintaining the same mean and std.dev. across tasks. It is not sufficient to mitigate forgetting. Figure 17 uses AE for reconstruction supplemented by an external VAE for generative replay. Though results are competitive with Flashcards, there is still forgetting in the previous tasks - MNIST and Fashion MNIST. Figure 18 presents results when us-

Figure 9. Different iterations of flashcards construction from Gaussian noise, after training autoencoder on SVHN

Figure 10. Different iterations of flashcards construction from new task data selected as Fashion MNIST, after training autoencoder on MNIST

Figure 11. Different iterations of flashcards construction from new task data selected as Cifar10, after training autoencoder on Fashion MNIST

Figure 12. Different iterations of flashcards construction from new task data selected as SVHN, after training autoencoder on Cifar10

ing Flashcards, where the past and current task samples are remembered well.

G. Sequence3 Continual Reconstruction

We also permuted the order by taking MNIST, Fashion MNIST and Cifar10, to further substantiate the effect of flashcards. Tables 4, 5 and 6 compare the mitigation of forgetting with flashcards.

H. Sequence5 Continual Denoising - Adjusting the weight of noise factor

We increased the noise factor steadily to check for the value where reconstruction fails completely. Figure 20 shows the impact of reconstruction using flashcards for different noise level settings. As more noise is added, it becomes visually difficult to make out the underlying image. At factor of 0.3, it is observed the network is trying to retain partial outer boundary but has forgotten the denoising ability when seeing the last task Omniglot.

I. Single Task - New Instance Classification Forgetting Setting

This section provides additional experiments run on Single Task, New-Instance Learning (ST-NIL) classification. Owing to page limitations, results are provided here. ST-NIL focuses on learning new instances every session while retaining the same number of classes. Performance indicates how well the model adapts to virtual concept drift across sessions. Flashcards are constructed from AE (unsupervised) per session and passed to the classifier to get predicted softmax scores as soft class-labels. It is observed that

Method	Task	MNIST	Fashion MNIST	Cifar10	Avg MAE
Joint Training	-	0.0141	0.0256	0.0629	0.0342
Coreset Sampling 5000	1	0.0190	-	-	0.0190
	2	0.0245	0.0388	-	0.0316
	3	0.0249	0.0395	0.0666	0.0436
Lower Bound	1	0.0190	-	-	0.0190
	2	0.0268	0.0268	-	0.0268
	3	0.0467	0.0469	0.0512	0.0482
Flashcards 5000	1	0.0190	-	-	0.0190
	2	0.0243	0.0310	-	0.0276
	3	0.0282	0.0366	0.0579	**0.0409**

Table 4. Sequence of 3 - Order: MNIST, Fashion MNIST, Cifar10 as 3 tasks. Tasks are added incrementally, and MAE is computed on each dataset after current task is completed.

Method	Task	Fashion MNIST	Cifar10	MNIST	Avg MAE
Joint Training	-	0.0141	0.0256	0.0629	0.0342
Coreset Sampling 5000	1	0.0324	-	-	0.0324
	2	0.0344	0.0589	-	0.0466
	3	0.0386	0.0661	0.0200	0.0415
Lower Bound	1	0.0324	-	-	0.0324
	2	0.0548	0.0564	-	0.0556
	3	0.0816	0.2996	0.0140	0.1317
Flashcards 5000	1	0.0324	-	-	0.0324
	2	0.0336	0.0520	-	0.0428
	3	0.0352	0.0637	0.0156	**0.0381**

Table 5. Sequence of 3 - Order: Fashion MNIST, Cifar10, MNIST as 3 tasks.

Method	Task	Cifar10	MNIST	Fashion MNIST	Avg MAE
Joint Training	-	0.0141	0.0256	0.0629	0.0342
Coreset Sampling 5000	1	0.0515	-	-	0.0515
	2	0.0639	0.0220	-	0.0429
	3	0.0654	0.0229	0.0336	**0.0406**
Lower Bound	1	0.0515	-	-	0.0515
	2	0.2602	0.0142	-	0.1372
	3	0.1233	0.0465	0.0371	0.0689
Flashcards 5000	1	0.0515	-	-	0.0515
	2	0.0625	0.0181	-	0.0403
	3	0.0664	0.0261	0.0308	0.0411

Table 6. Sequence of 3 - Order: Cifar10, MNIST, Fashion MNIST as 3 tasks.

Session/ Method	Type	1	2	3	4	5
Naive*	-	67.80	69.31	71.37	73.12	73.23
Cumulative*	-	67.80	76.13	81.22	81.83	82.12
EEIL 1K*	ER	67.80	71.97	73.27	74.91	74.66
A-GEM 1K*	ER	67.80	72.27	73.72	74.81	75.15
EWC*	Reg	67.80	69.45	72.68	74.02	74.31
SI*	Reg	67.80	70.48	72.82	74.63	74.58
IMM*	Reg	67.80	69.69	72.85	74.37	73.84
Flashcards1K	FR	67.90	71.40	73.34	74.84	74.88

Table 7. ST-NIL Classification on Cifar10. * are reported from the paper. Flashcards created from unsupervised AE performs equally well in comparison to other methods primarily built for classification. Reg=Regularization, ER=Episodic Replay, FR=Flashcard Replay.

flashcards' performance is better than regularization and on-par with episodic replay, without explicitly storing exemplars in memory. (Table 7). ResNet18 is used as classifier, optimized using SGD with learning rates of 0.001 over 20

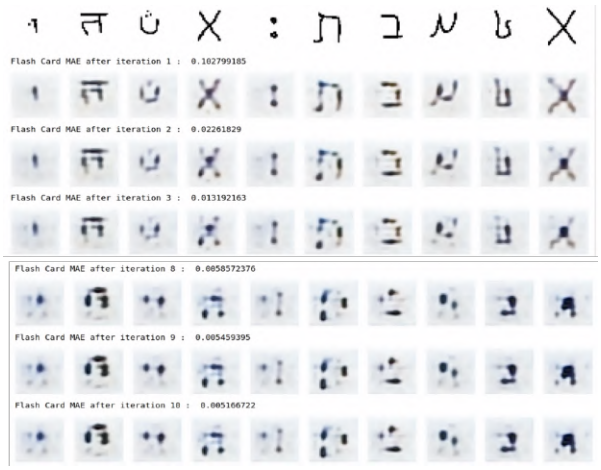


Figure 13. Different iterations of flashcards construction from new task data selected as Omniglot, after training autoencoder on SVHN

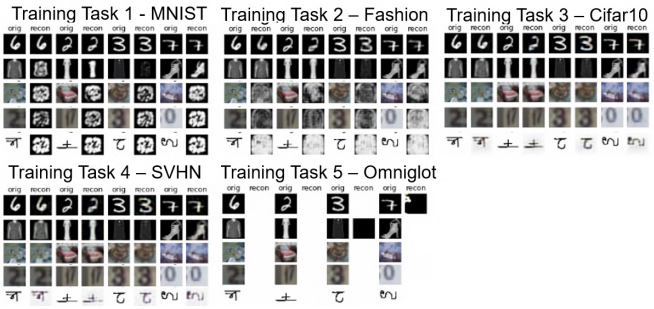


Figure 14. Continual Reconstruction on Naive / Sequential Fine Tuning (SFT).

Figure 15. Continual Reconstruction using episodic memory - coreset 500.

epochs, and new sessions are introduced by adopting the same brightness and saturation from the paper , test set is constant across sessions.

Figure 16. Continual Reconstruction using VAE trained exlusively for Continual Learning.

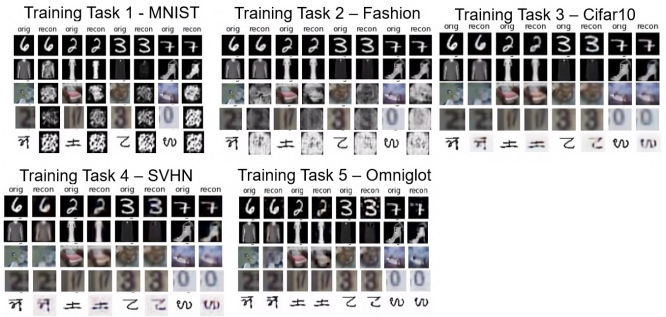
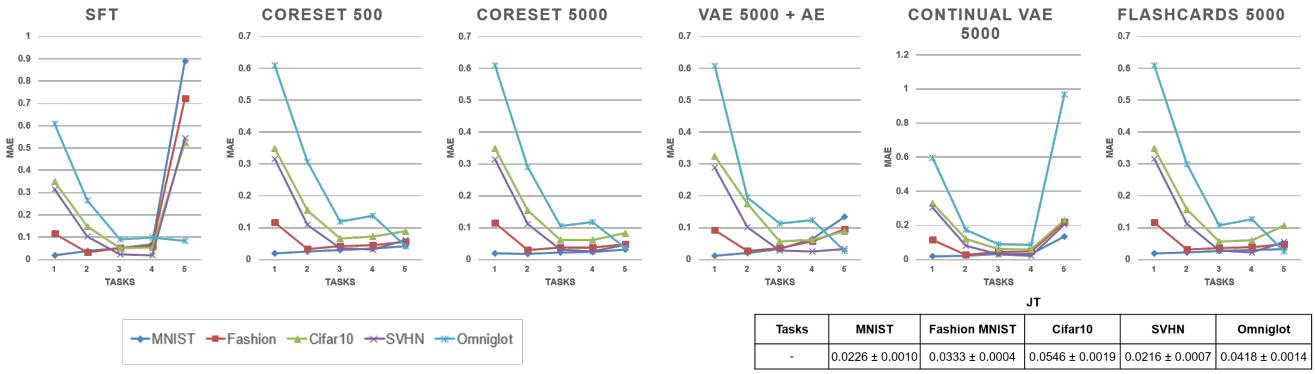


Figure 17. Continual Reconstruction using AE + VAE as generative replay.

Figure 18. Continual Reconstruction using Flashcards.



JT

Tasks	MNIST	Fashion MNIST	Cifar10	SVHN	Omniglot
-	0.0226 ± 0.0010	0.0333 ± 0.0004	0.0546 ± 0.0019	0.0216 ± 0.0007	0.0418 ± 0.0014

Figure 19. Continual Learning for Reconstruction. Individual graphs for different methods show the variation of test Mean Absolute Error (MAE) on current task dataset after observing the data for sequence of tasks. The table shows the test MAE for Joint Training (JT) method. The reported values in graph and table obtained over 5 experimental runs. The standard deviation is quite small and is not displayed on the graphs to avoid clutter.

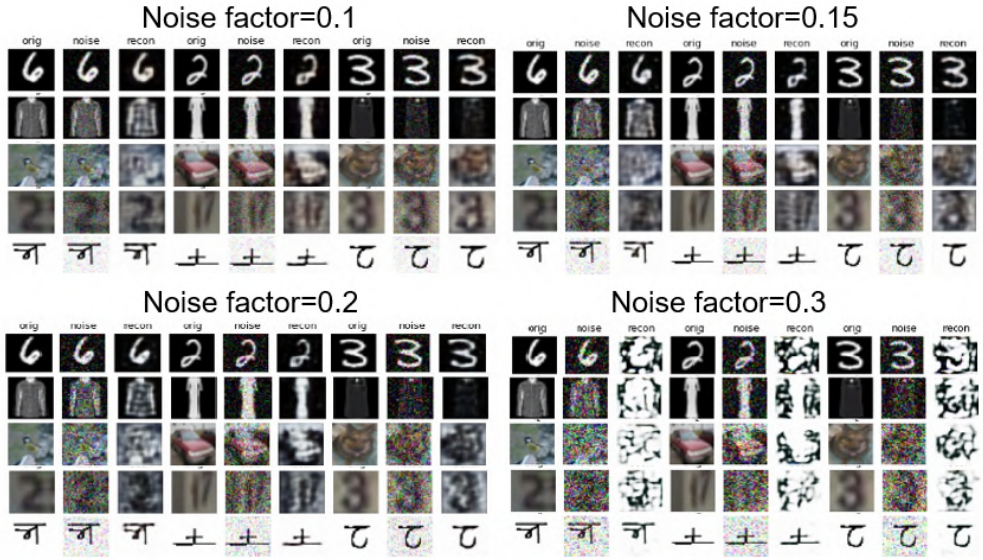


Figure 20. Continual Denoising scenario. Shown in figure is the effect of noise applied and the reconstruction of Sequence5 using flashcards.