(a) Class 0 (b) Class 1

Fig. 3. A sample from each class of the toy classification dataset.

APPENDIX

This section provides additional details on the (semi-) synthetic data used in Section V.

A. Toy Classification

In correspondence to [27], we create a toy dataset of $5\times5\times3$ RGB images with four possible colors. Created images fall into two classes with two independent decision rules that a model could implicitly learn: whether the four corner pixels are of the same color, and whether their top-middle three pixels are all different colors. Images in class 1 satisfy both conditions and images in class 2 satisfy neither. As this dataset does not contain multiple classes or domains, it is only used as a sanity check in Section V-A.

B. Decoy and Split Decoy MNIST

Inspired by the popular continuous learning benchmarking dataset Split MNIST [47], we apply the principle to Decoy MNIST to generate ground truth data and a rule that we can apply to interact with the model. On Decoy MNIST [27], images x have 4×4 gray swatches in randomly chosen corners whose shades are functions of their digits y in training (in particular, $255-25y$) but are random in test. In correspondence to Split MNIST, we split the items of Decoy MNIST into 5 tasks containing two consecutive classes (see Figure 4).

C. Tabular and Continous Tabular

Similar to [48], we generate a tabular dataset with 10 classes and 20 features, of which 10 are informative using the scikit-learn implementation[3]. Therefore, the remaining features are irrelevant and should not be used in the classifier. The approach generates clusters of normally distributed points about vertices of a 10-dimensional hypercube with sides of length 20 and assigns an equal number of clusters to each class. It introduces interdependence between these features and adds various noise to the data. Similar to Decoy MNIST, we split the tabular data into 5 consecutive tasks based on the label for the continuous experiments.

[3]https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html

D. Time Series and Continuous Time Series

Inspired by [49], we generate data for 50 time steps with a feature size of 1 from 5 different processes. For each process, we divide the generated data into two classes by adding or subtracting a constant ϵ in the area of the desired informative features, resulting in a 10 classes classification problem (5 processes x 2 classes per process). The data is generated based on 5 time processes with $\epsilon_t \sim N(0,1)$ and the informative features are in the middle of the time series, masking 35% of the time steps.

- Gaussian ($\mu = 0, \sigma = 0$):
  $X_t = \epsilon_t$
- Harmonic:
  $X(t) = sin(2\pi 2t) + e_t$
- Pseudo Periodic ($A_t \sim N(0, 0.5)$, $f_t \sim N(2, 0.01)$):
  $X(t) = A_t sin(2\pi f_t, t) + \epsilon_t$
- Autoregressive ($p = 1$, $\varphi = 0.9$):
  $X_t = \sum_{i=1}^{p} \varphi X_{t-i} + \epsilon_t$ $X_t = \varphi X_{t-1} + \sigma(1-\varphi)^2 * \epsilon + \epsilon_t$
- NARMA ($n = 10$, $U \sim U(0.05)$):
  $X_t = 0.3X_{t-1} + 0.05X_{t-1}\sum_{i=0}^{n-1} X_{t-1} + 1.5U(t-(n-1)) * U(t) + 0.1 + \epsilon_t$

For the Finetuning all types of generation processes are in the initial training data shuffled. For the Continuous setting, leaned onto domain incremental learning, we split the dataset into tasks according to the underlying time series process.

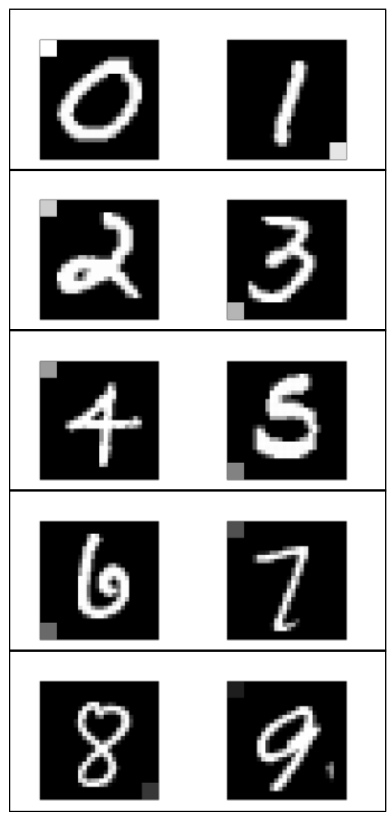

Fig. 4. In the Split Decoy MNIST 5 tasks need to be learned consecutively without forgetting the previously learned tasks.