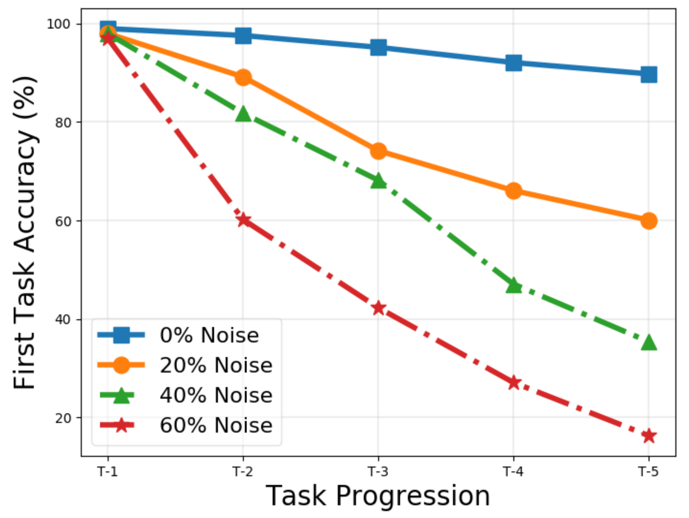
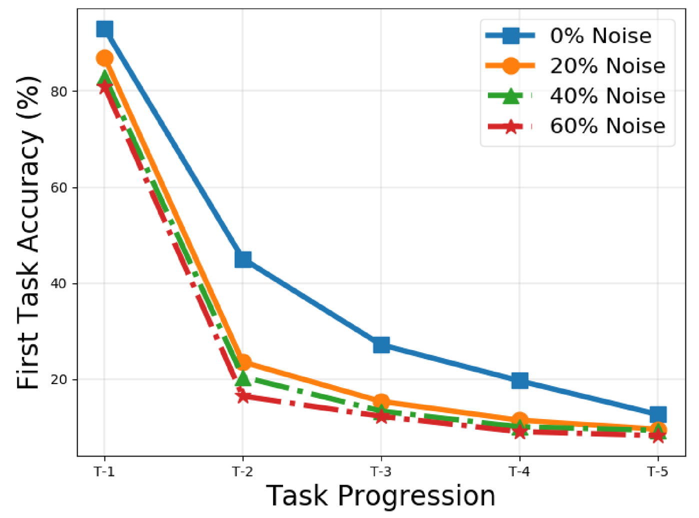
CNLL: A Semi-supervised Approach For Continual Noisy Label Learning

Nazmul Karim (nazmul.karim18@knights.ucf.edu), Umar Khalid, Ashkan Esmaeili, and Nazanin Rahnavard
University of Central Florida

Overview

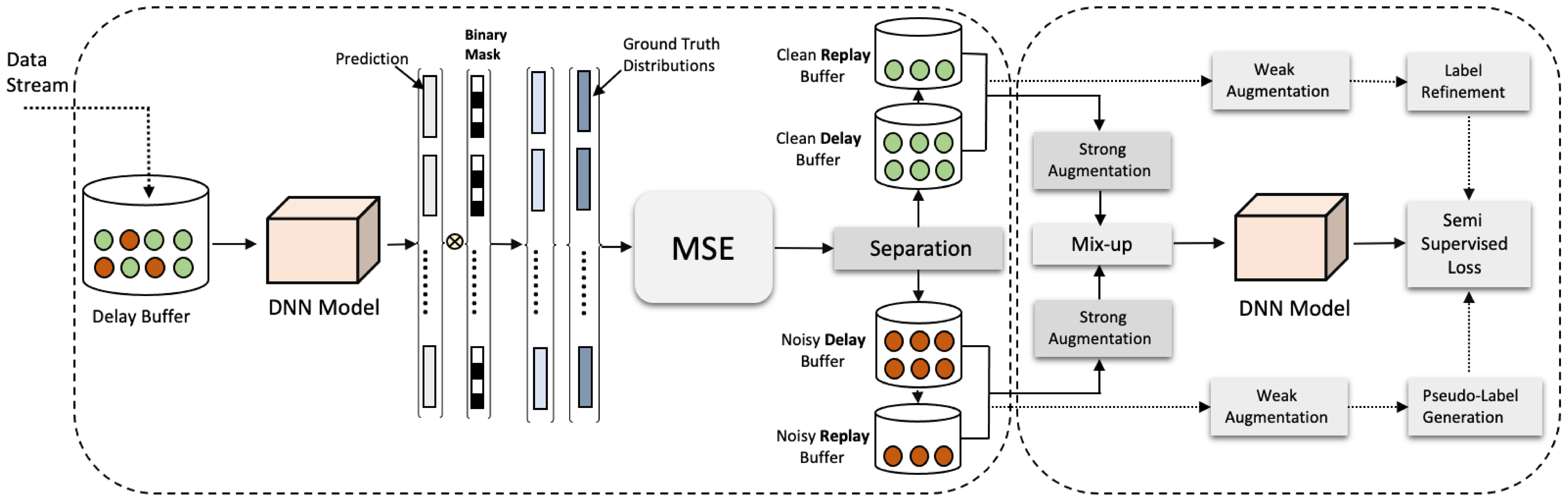
- **Goal:** Continual learning under noisy data stream.
- **Main characteristics of CNLL:**
  - Loss-based separation of the clean and noisy data stream.
  - Mask-based implementation for filtering the classes that are only present in the data stream.
  - Semi-supervised learning for separated stream.





Proposed Algorithm: CNLL

- Consider a scenario where a model is trained on continuous data stream that may contain noisy labels.
- Based on the loss-values of all samples in the current data stream, a suitable threshold has been used to separate the data stream into clean and noisy delay buffer.
- Instead of using a single replay buffer, a dual-buffer system has been implemented where a portion of samples from the delay buffer is stored.
- A specific data-augmentation technique has been implemented to train the model on both the clean and noisy data streams.

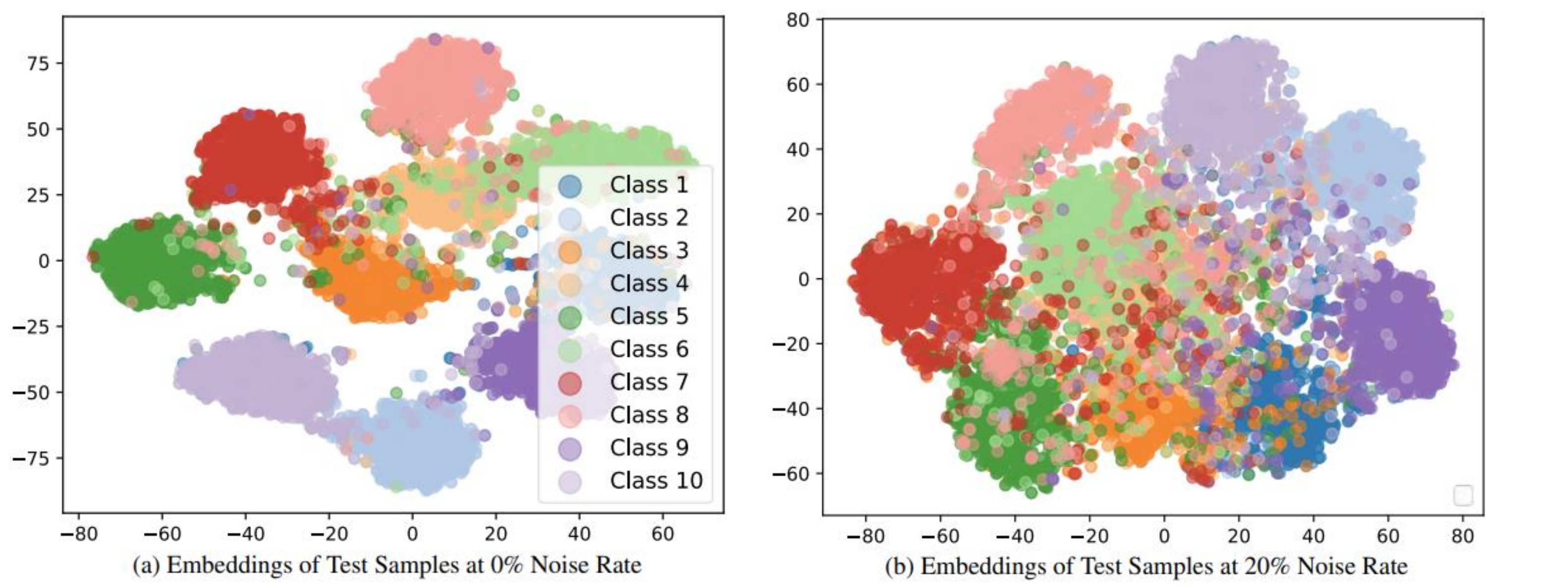


Separation Algorithm

Algorithm 1 Task-free Sample Separation for CNLL

Input: Training data $(\mathbf{x}_t, y_t), \ldots, (\mathbf{x}_{T_{train}}, y_{T_{train}})$, Warmup Period E, and network parameters Θ.
$\mathcal{C} = \mathcal{N} = \{\}$ *// Initialize Clean and Noisy Delay Buffer*
$\mathcal{B}_\mathcal{C} = \mathcal{B}_\mathcal{N} = \{\}$ *// Initialize Clean and Noisy Replay Buffer*
$\mathcal{D} = \{\}$ *// Initialize Delay buffer*
for $t = 1$ **to** T_{train} **do**
  if $\mathcal{D}$ is full **then**
    $\Theta \leftarrow \textsc{WarmUp}(\Theta, \mathcal{D}, E)$
    for $i = 1$ **to** $|\mathcal{D}|$ **do**
      $\mathbf{p}_i = \mathbf{f}(\mathbf{x}_i; \Theta)$
      Initialize binary mask $\mathbf{m}_i$
      $l_i = MSE(\mathbf{y}_i, \mathbf{m}_i \odot \mathbf{p}_i)$
    end for
    Calculate $l_{threshold}$ using Eq. 3
    $\mathcal{D}_{clean} \leftarrow \{(x_i, y_i) : \forall\ l_i < l_{threshold}\}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{D}_{clean}$
    $\mathcal{D}_{noisy} \leftarrow \mathcal{D} \setminus \mathcal{D}_{clean}$
    $\mathcal{N} \leftarrow \mathcal{N} \cup \mathcal{D}_{noisy}$
    Reset $\mathcal{D}$
  else
    update $\mathcal{D}$ with $(\mathbf{x}_t, y_t)$
  end if
  if $\mathcal{C}$ is full **then**
    Update $\mathcal{B}_\mathcal{C}$ using eq. 4
    Reset $\mathcal{C}$
  end if
  if $\mathcal{N}$ is full **then**
    Update $\mathcal{B}_\mathcal{N}$ using eq. 5
    Reset $\mathcal{N}$
  end if
  if $\mathcal{C}$ and $\mathcal{N}$ is full **then**
    Fine-tune and Inference Phase for Θ
  end if
end for

T-SNE Visualization



(a) Embeddings of Test Samples at 0% Noise Rate
(b) Embeddings of Test Samples at 20% Noise Rate

CNLL achieves good classification performance with and without noisy labels.

Experimental Results

	MNIST					CIFAR-10				
	symmetric			asymmetric		symmetric			asymmetric	
Noise rate (%)	20	40	60	20	40	20	40	60	20	40
Multitask 0% noise [9]	98.6					84.7				
Finetune	19.3	19.0	18.7	21.1	21.1	18.5	18.1	17.0	15.3	12.4
EWC [30]	19.2	19.2	19.0	21.6	21.1	18.4	17.9	15.7	13.9	11.0
CRS [66]	58.6	41.8	27.2	72.3	64.2	19.6	18.5	16.8	28.9	25.2
CRS + L2R [51]	80.6	72.9	60.3	83.8	77.5	29.3	22.7	16.5	39.2	35.2
CRS + Pencil [72]	67.4	46.0	23.6	72.4	66.6	23.0	19.3	17.5	36.2	29.7
CRS + SL [67]	69.0	54.0	30.9	72.4	64.7	20.0	18.8	17.5	32.4	26.4
CRS + JoCoR [68]	58.9	42.1	30.2	73.0	63.2	19.4	18.6	21.1	30.2	25.1
PRS [26]	55.5	40.2	28.5	71.5	65.6	19.1	18.5	16.7	25.6	21.6
PRS + L2R [51]	79.4	67.2	52.8	82.0	77.8	30.1	21.9	16.2	35.9	32.6
PRS + Pencil [72]	62.2	33.2	21.0	68.6	61.9	19.8	18.3	17.6	29.0	26.7
PRS + SL [67]	66.7	45.9	29.8	73.4	63.3	20.1	18.8	17.0	29.6	24.0
PRS + JoCoR [68]	56.0	38.5	27.2	72.7	65.5	19.9	18.6	16.9	28.4	21.9
MIR [2]	57.9	45.6	30.9	73.1	65.7	19.6	18.6	16.4	26.4	22.1
MIR + L2R [51]	78.1	69.7	49.3	79.4	73.4	28.2	20.0	15.6	35.1	34.2
MIR + Pencil [72]	70.7	34.3	19.8	79.0	58.6	22.9	20.4	17.7	35.0	30.8
MIR + SL [67]	67.3	55.5	38.5	74.3	66.5	20.7	19.0	16.8	28.1	22.9
MIR + JoCoR [68]	60.5	45.0	32.8	72.6	64.2	19.6	18.4	17.0	27.6	23.5
GDumb [47]	70.0	51.5	36.0	78.3	71.7	29.2	22.0	16.2	33.0	32.5
GDumb + L2R [51]	65.2	57.7	42.3	67.0	62.3	28.2	25.5	18.8	30.5	30.4
GDumb + Pencil [72]	68.3	51.6	36.7	78.2	70.0	26.9	22.3	16.5	32.5	29.7
GDumb + SL [67]	66.7	48.6	27.7	73.4	68.1	28.1	21.4	16.3	32.7	31.8
GDumb + JoCoR [68]	70.1	56.9	37.4	77.8	70.8	26.3	20.9	15.0	33.1	32.2
SPR [27]	85.4	86.7	84.8	86.8	86.0	43.9	43.0	40.0	44.5	43.9
CNLL(ours)	**92.8**	**90.1**	**88.8**	**91.5**	**89.4**	**68.7**	**65.1**	**52.8**	**67.2**	**59.3**

We achieve around 4% accuracy improvement for the MNIST dataset and around 10% improvement for the CIFAR10 dataset. The reported accuracy is the average accuracy on all tasks combined.

CWNLab

Codebase