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 \((x_1, y_1), ..., (x_{T_{train}}, y_{T_{train}})\).

- **Warmup Period** \(E\), and network parameters \(\theta\).
- **Delay Buffer** \(\mathcal{D}_B\).

**Output:** \(\mathcal{C} = \{C\}\) // Initialize Clean and Noisy Delay Buffer.

1. **Initialize** \(\mathcal{D}_C = \{C\}\) // Initialize Clean and Noisy Replay Buffer.
2. **D =** \(\{C\}\) // Initialize Delay buffer.
3. **for** \(i = 1 \text{ to } T_{train}\) **do**
   - **if** \(D\) is full then
     - \(\theta \leftarrow \text{Warmup}(\theta, D, E)\)
   - **for** \(j = 1 \text{ to } D\) **do**
     - \(p_j \leftarrow f(x_j, \theta)\)
     - Initialize binary mask \(m_j\)
     - \(l_j = \text{MSE}(y_j, m_j \odot p_j)\)
     - **end**
     - **Calculate** \(t_{threshold} \text{ using Eq. 3}\)
     - \(D_C \leftarrow \{x_j | y_j \notin t_{threshold}\}\)
     - \(D_N \leftarrow \{x_j | y_j \in t_{threshold}\}\)
     - **else**
       - **end**
       - **end**
       - **end**
     - **if** \(C\) is full then
       - **Update** \(\mathcal{D}_B\) by using eq. 4
       - **reset** \(C\)
     - **else**
       - **if** \(N\) is full then
         - **Update** \(\mathcal{D}_B\) by using eq. 5
         - **reset** \(N\)
       - **end**
       - **if** \(C\) and \(N\) is full then
         - **Fine-tune and Inference Phase for \(\theta\)**
       - **end**
     - **end**
   - **end**
4. **end**

Experimental Results

**T-SNE Visualization**

CNLL achieves good classification performance with and without noisy labels.

**Conclusion**

We achieve around 4% accuracy improvement for the MNIST dataset and around 16% improvement for the CIFAR10 dataset. The reported accuracy is the average accuracy on all tasks combined.