Task-agnostic Continual Learning with Hybrid Probabilistic Models

Polina Kirichenko [1] Mehrdad Farajtabar [2] Dushyant Rao [2]
Balaji Lakshminarayanan [3] Nir Levine [2] Ang Li [2]
Huiyi Hu [2] Andrew Gordon Wilson [1] Razvan Pascanu [2]

[1]New York University

[2]DeepMind

[3]Google Brain

August 16, 2021

Outline

Existing Approaches

- re-sample the data or design specific loss functions that better facilitate learning with imbalanced data
- enhance recognition performance of the tail classes by transferring knowledge from the head classes

Our Contribution

- Hybrid Continual Learning (HCL) - a normalizing flow-based approach.
- Generative replay and a novel functional regularization are employed to alleviate forgetting. The functional regularization is shown to be better than generative replay.
- HCL achieves strong performance on *split MNIST*, *split CIFAR*, *SVHN-MNIST* and *MNIST-SVHN* datasets.
- HCL can detect task boundaries and identify new as well as recurring tasks.

Continual Learning (CL)

- A CL model $g_\theta : \mathcal{X} \to \mathcal{Y}$.
- A sequence of τ supervised tasks: $T_{t_1},\ T_{t_2}, \ldots, T_{t_\tau}$. τ is not known in advance.
- Each task $T_i = \{(x_j^i, y_j^i)\}_{j=1}^{N_i}$, where $x_j^i \in \mathcal{X}^i$ and $y_j^i \in \mathcal{Y}^i$.
- The corresponding data distribution of task T_i is $p_i(x, y)$.
- **Constraint**: While training on a task T_i the model cannot access to the data from previous $T_1, \ldots, T_{i-1}$ or future tasks $T_{i+1}, \ldots, T_\tau$.
- **Objective**: Minimize $\sum_{i=1}^{M} \mathbb{E}_{x,y\sim p_i(\cdot,\cdot)} l(g_\theta(x), y)$ for some risk function $l(\cdot,\cdot)$ and generalize well on all tasks after training.

Modeling the Data Distribution

- $p_t(x, y)$: the joint distribution of the data x and the class label y conditioned on a task t.

$$p_t(x, y) \approx \hat{p}(x, y|t) = \hat{p}_X(x|y, t)\hat{p}(y|t)$$

- $\hat{p}_X(x|y, t)$ is modeled by a normalizing flow f_θ with a base distribution $\hat{p}_Z = \mathcal{N}(\mu_{y,t}, I)$.

$$\hat{p}_X(x|y, t) = f_\theta^{-1}\left(\mathcal{N}(\mu_{y,t}, I)\right)$$

- $\mu_{y,t}$ is the mean of the latent distribution corresponding to the class y and task t.
- $\hat{p}(y|t)$ is assumed to be a uniform distribution over the classes for each task: $\hat{p}(y|t) = 1/K$.

Task Identification

- log-likelihood

$$S_1(B, t) = \sum_{(x_j, y_j) \in B} \hat{p}_X(x_j | y_j, t)$$

- log-likelihood of the latent variable

$$S_2(B, t) = \sum_{(x_j, y_j) \in B} \hat{p}_Z(f_\theta(x_j) | y_j, t)$$

- log-determinant of the Jacobian

$$S_3(B, t) = S_1(B, t) - S_2(B, t)$$

Generative Replay (HCL-GR)

- Store a single snapshot of the HCL model $\hat{p}_X^{(k)}(x|y,t)$ with weights $\theta^{(k)}$.
- Generate and replay data from old tasks using the snapshot: $x_{GR} \sim \hat{p}_X^{(k)}(x|y,t)$, where $y \sim U\{1,\dots,K\}$ and $t \sim U\{t_1,\dots,t_k\}$.
- Maximize the likelihood $\mathcal{L}_{GR} = \log \hat{p}_X(x_{GR}|y,t)$ under the current model $\hat{p}_X(\cdot)$.
- The resulting loss in generative replay training is $\mathcal{L}_{ll} + \mathcal{L}_{GR}$, where $\mathcal{L}_{ll}$ is the log-likelihood of the data on the current task.
- Update the snapshot with new weights $\theta^{(k+1)}$ after detecting the task change $T_{t_{k+1}} \to T_{t_{k+2}}$.

Functional Regularization (HCL-FR)

Enforce the flow to map samples from previous tasks to the same latent representations as a snapshot model.

- Store a single snapshot of the model $\hat{p}_X^{(k)}(\cdot)$ and produce samples $x_{FR} \sim \hat{p}_X^{(k)}(x|y,t)$ for $y \sim U\{1,\ldots,K\}$, $t \sim U\{t_1,\ldots,t_k\}$.
- $\mathcal{L}_{FR} = \|f_\theta(x_{FR}) - f_{\theta^{(k)}}(x_{FR})\|^2$, where f_θ is the current flow mapping and $f_{\theta^{(k)}}$ is the snapshot model.
- The resulting loss in functional regularization is $\mathcal{L}_{ll} + \alpha\mathcal{L}_{FR}$.

Theoretical Analysis

Compared Methods

- Adam: Train the model with Adam optimizer without any extra steps for preventing forgetting.
- Multi-Task Learning (MTL): When training on T_{t_i}, it has access to all previous tasks $T_{t_1}, \ldots, T_{t_{i-1}}$.
- Experience Replay (ER):
- CURL:

Datasets

- Split MNIST: Split the dataset into 5 binary classification tasks.
- MNIST-SVHN and SVHN-MNIST:
- Split CIFAR:

Experiment Results

Experiment Results on Split CIFAR