Positive-Unlabeled Compression on the Cloud

Yixing Xu[1], Yunhe Wang[1], Hanting Chen[2], Kai Han[1], Chunjing Xu[1], Dacheng Tao[3], Chang Xu[3]

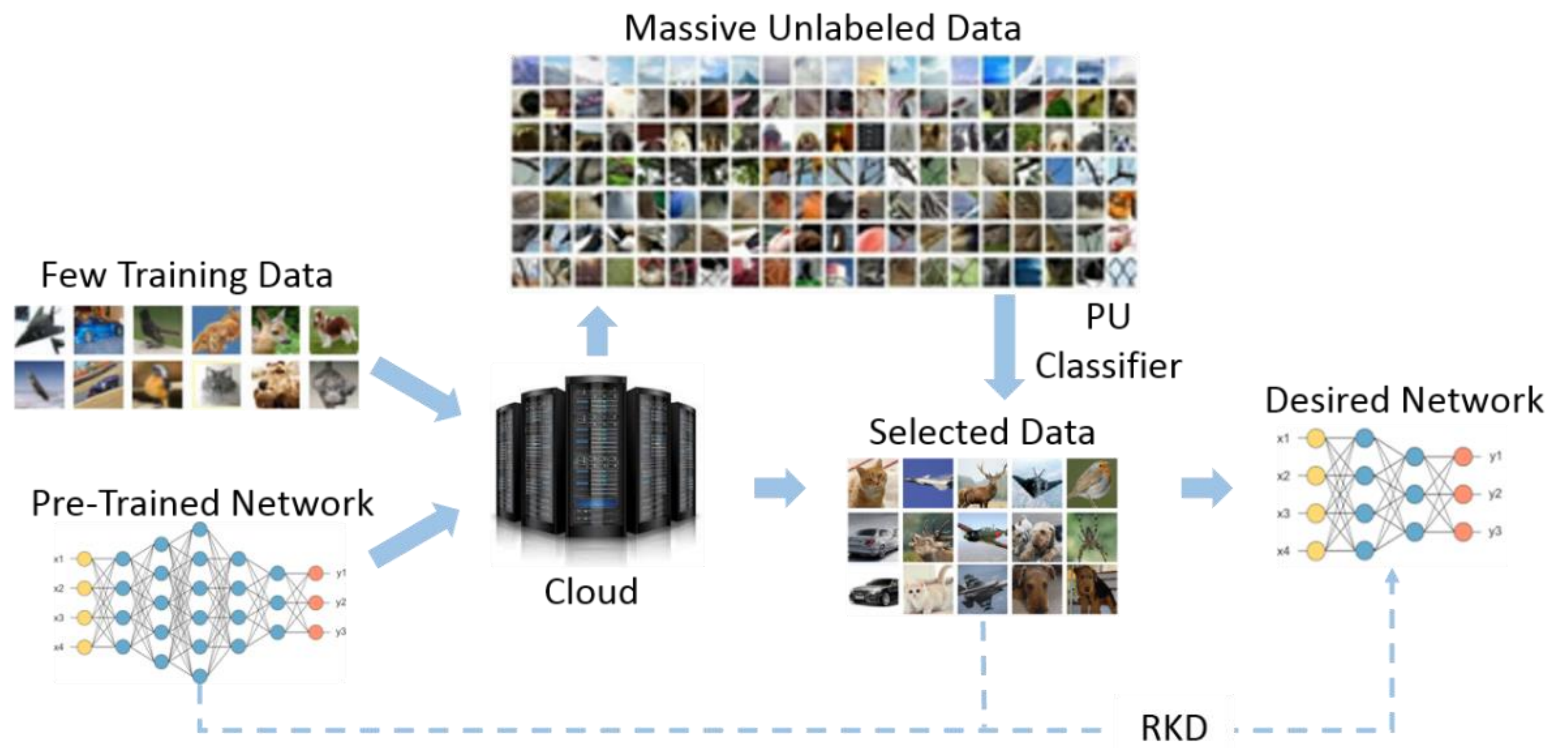
[1]Huawei Noah's Ark Lab; [2]Peking University; [3]The University of Sydney

Abstract

Many attempts have been done to extend the great success of convolutional neural networks (CNNs) achieved on high-end GPU servers to portable devices such as smart phones. Providing compression and acceleration service of deep learning models on the cloud is therefore of significance and is attractive for end users. However, existing network compression and acceleration approaches usually fine-tuning the svelte model by requesting the entire original training data (e.g. ImageNet), which could be more cumbersome than the network itself and cannot be easily uploaded to the cloud. In this paper, we present a novel positive-unlabeled (PU) setting for addressing this problem. In practice, only a small portion of the original training set is required as positive examples and more useful training examples can be obtained from the massive unlabeled data on the cloud through a PU classifier with an attention based multi-scale feature extractor. We further introduce a robust knowledge distillation (RKD) scheme to deal with the class imbalance problem of these newly augmented training examples. The superiority of the proposed method is verified through experiments conducted on the benchmark models and datasets. We can use only 8% of uniformly selected data from the ImageNet to obtain an efficient model with comparable performance to the baseline ResNet-34.

Motivation

- Providing compression service on the cloud.
- Little training data can be obtained due to privacy reason.

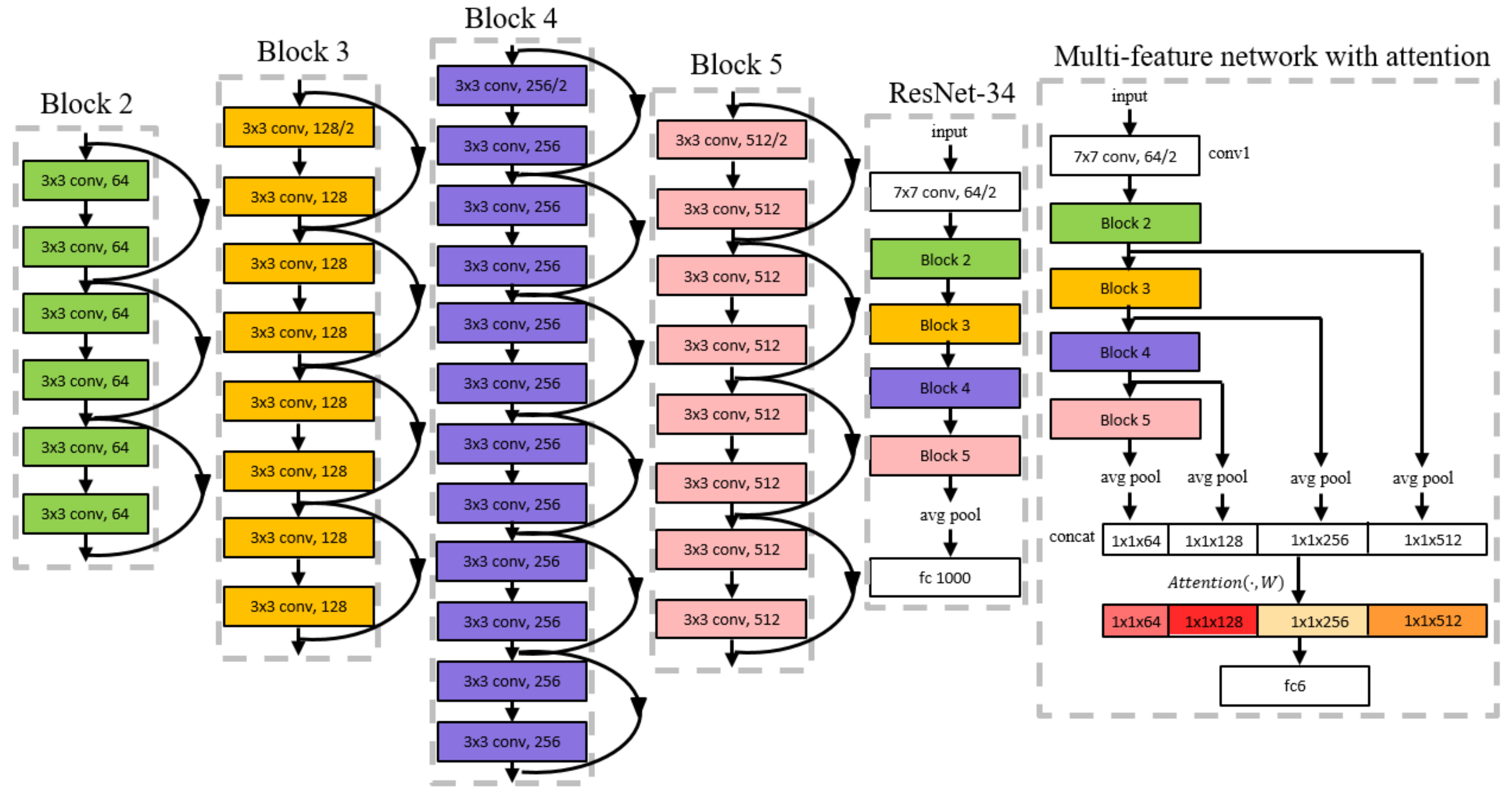


Positive-Unlabeled Classifier for Selecting Data

Non-negative PU loss function:

$$\tilde{R}_{pu}(f) = \pi_p \hat{R}_p^+(f) + \max\{0, \hat{R}_{\mathbf{x}}(f) - \pi_p \hat{R}_p^-(f)\} \quad (3)$$

Attention based multi-scale feature extractor:



```
Algorithm 1 PU classifier for more data.
Require: An initialized network $\mathcal{N}_A$, a tiny labeled dataset $L^t$ and an unlabeled dataset $U$.
1: Module 1: Train PU Classifier
2: repeat
3:     Randomly select a batch $\{\mathbf{x}_i^U\}_{i=1}^N$ from $U$ and $\left\{\mathbf{x}_i^{L^t}\right\}_{i=1}^N$ from $L^t$;
4:     Optimize $\mathcal{N}_A$ following Eq. 3;
5: until convergence
6: Module 2: Extend the labeled dataset
7: Obtain the positive data $U^p$ from $U$ utilizing PU classifier $\mathcal{N}_A$;
8: Unify the positive dataset $U^p$ and tiny dataset $L^l$ to achieve extended dataset $L^l = L^t \cup U^p$;
Ensure: Extended dataset $L^l$.
```

Robust Knowledge Distillation

- Data obtained from PU method suffer from noise and data imbalanced problem.
- Assign weight to each category of the samples:

$$w_{kd}^k = \frac{K/y^k}{\sum_{k=1}^K 1/y^k}, \quad k = 1, 2, \cdots, K \quad (7)$$

- The surrogate KD loss dealing with imbalanced problem:

$$\tilde{\mathcal{L}}^{KD} = \frac{1}{n}\sum_i w_i \mathcal{F}_{ce}(\mathbf{y}_i^{te}, \mathbf{y}_i^{st}) \quad (8)$$

- Perturbations of the original weights:

$$p(|w_{kd}^k - w_{kd}^{*k}| < \epsilon) > 1 - \delta \quad (9)$$

- Optimization function dealing with noise:

$$\mathcal{N}_{st}^{\mathbf{W}} = \arg\min_{\mathcal{N}_{st}\in\mathcal{N}} \max_{\mathbf{w}\in\mathbf{W}} \tilde{\mathcal{L}}^{KD}(\mathcal{N}_{st}, \mathbf{w}) \quad (10)$$

```
Algorithm 2 Robust Knowledge distillation.
Require: A given teacher network $\mathcal{N}_{te}$, the extended dataset $L^l$ and a hyper-parameter $\epsilon$.
1: Initialize the student network $\mathcal{N}_{st}$;
2: Calculate weight vectors $\mathbf{w}_{kd}$ using Eq. 7 and generative a set $\mathbf{W}$ using a random perturb $\epsilon$;
3: repeat
4:     Randomly select a batch $\left\{\mathbf{x}_i^{L^l}\right\}_{i=1}^m$;
5:     Employ the teacher and student network: $\mathbf{y}_i^{te} \leftarrow \mathcal{N}_{te}(\mathbf{x}_i^{L^l})$; $\mathbf{y}_i^{st} \leftarrow \mathcal{N}_{st}(\mathbf{x}_i^{L^l})$
6:     Calculate the surrogate KD loss $\tilde{\mathcal{L}}^{KD}$ following Eq. 8;
7:     Update $\mathcal{N}_{st}^{\mathbf{W}}$ with Eq. 10;
8: until convergence
Ensure: The student network $\mathcal{N}_{st}$.
```

Experiments

We tested the performance of the proposed method on CIFAR-10, ImageNet and MNIST datasets.

Table 1: Classification results on CIFAR-10 dataset. The best results are bold in the table.

Method	n_l	n_t	Data source	FLOPs	#params	Acc(%)
Teacher	-	50,000	Original Data	1.16G	21M	95.61
KD [9]	-	50,000	Original Data	557M	11M	94.40
Baseline-1	-	269,427	Manually selected data	557M	11M	93.44
Baseline-2	-	50,000	Randomly selected data	557M	11M	87.02
PU-s1	100	110,608	PU data	557M	11M	**93.75**
	50	94,803				93.02
	20	74,663				92.23
PU-s2	100	50,000	PU data	557M	11M	91.56
	50	50,000				91.33
	20	50,000				91.27

Table 2: Classification results on ImageNet dataset. "KD-all" utilizes the entire ImageNet training dataset to train the student network. "KD-500k" randomly selects 500k training data from ImageNet for learning the student network.

Algorithm	n_t	Data source	FLOPs	#params	top-1 acc(%)	top-5 acc(%)
Teacher	1,281,167	Original Data	3.67G	22M	73.27	91.26
KD-all	1,281,167	Original Data	1.82G	12M	68.67	88.76
KD-500k	500,000	Original Data	1.82G	12M	63.90	85.88
PU-s1	690,978	PU data	1.82G	12M	61.92	86.00
PU-s2	500,000	PU data	1.82G	12M	61.21	85.33

Table 3: Comparsion on the state-of-the-art methods on the MNIST dataset.

	1	2	5	10	20	all-meta-data
data-free KD [16]	-	-	-	-	-	92.5
FitNet [20]	90.3	94.2	96.1	96.7	97.3	-
FSKD [15]	95.5	97.2	97.6	98.0	98.1	-
PU-s1	**98.5**	**98.7**	**98.7**	**98.8**	**98.9**	-
PU-s2	98.3	98.5	98.5	98.6	98.6	-

- Teacher: ResNet-34 + SGD
- Student: ResNet-18 + SGD
- KD: ResNet-18 + standard KD method
- Baseline-1: ResNet-18 + manual select data+ proposed
- Baseline-2: ResNet-18 + random select data+ proposed
- PU-s1: ResNet-18 + all PU data + proposed
- PU-s2: ResNet-18 + limited PU data + proposed

References

- R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama. Positive-unlabeled learning with nonnegative risk estimator. NIPS, 2017
- G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. NIPS workshop, 2014.
- J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. CVPR, 2018.
- R. Wang and K. Tang. Minimax classifier for uncertain costs. arXiv, 2012.
- T. Li, J. Li, Z. Liu, and C. Zhang. Knowledge distillation from few samples. arXiv, 2018.
- H.Chen, Y.Wang, C.Xu, Z.Yang, C.Liu, B.Shi, C.Xu, C.Xu and Q.Tian. Data-Free Learning of Student Networks. ICCV, 2019.