Comparison of Deep Co-Training and Mean-Teacher approaches for semi-supervised audio tagging

Léo Cances, Thomas Pellegrini

COMPARISON OF DEEP CO-TRAINING AND MEAN-TEACHER APPROACHES FOR SEMI-SUPERVISED AUDIO TAGGING

Léo Cances, Thomas Pellegrini

IRIT, Université Paul Sabatier, CNRS, Toulouse, France

ABSTRACT

Recently, a number of semi-supervised learning (SSL) methods, in the framework of deep learning (DL), were shown to achieve state-of-the-art results on image datasets, while using a (very) limited amount of labeled data. To our knowledge, these approaches adapted and applied to audio data are still sparse, in particular for audio tagging (AT). In this work, we adapted the Deep-Co-Training algorithm (DCT) to perform AT, and compared it to another SSL approach called Mean Teacher (MT), that has been used by the winning participants of the DCASE competitions these last two years. Experiments were performed on three standard audio datasets: Environmental Sound classification (ESC-10), UrbanSound8K, and Google Speech Commands. We show that both DCT and MT achieved performance approaching that of a fully supervised training setting, while using a fraction of the labeled data available, and the remaining data as unlabeled data. In some cases, DCT even reached the best accuracy, for instance, 72.6% using half of the labeled data, compared to 74.4% using all the labeled data. DCT also consistently outperformed MT in almost all configurations. For instance, the most significant relative gains brought by DCT reached 12.2% on ESC-10, compared to 7.6% with MT. Our code is available online[1].

***Index Terms*—** Audio tagging, semi-supervised learning, deep co-training, mean-teacher

1. INTRODUCTION

Semi-supervised learning (SSL) approaches utilize a set of labeled data and a larger set of unlabeled data that are cheaper and faster to obtain. For audio event classification [1, 2] and for speech recognition [3], the most straightforward approach is pseudo-labeling. It consists of an iterative process that starts by training a system on the available labeled set, and then making predictions on the unlabeled set. Labels with the most confidence are kept and added to the labeled set. The system is retrained, and the whole process can be repeated for several iterations until the gain in performance is marginal.

A number of better approaches have been proposed since then, for instance Mean Teacher (MT) [4, 5], originally tested on image datasets. We can find audio applications of MT in the Detection and Classification of Acoustic Scenes and Events (DCASE) 2018 and 2019 task 4 challenges, namely the weakly supervised Sound Event Detection task. The 2018 winners trained convolutional neural networks (CNN) on both a small labeled subset and a larger unlabeled one [5]. This approach consists of training a first model ("student") to make predictions consistent to those of a second model ("teacher"). This is achieved by a consistency cost computed on the predictions made by the two models on the unlabeled subset. The teacher model receives the same input as the student model but slightly perturbed, though, to ensure a virtuous collaboration between the models, and to avoid collapsing the two models.

Deep-Co-Training (DCT) is a deep learning method inspired by the reknown Co-Training framework [6]. It has been successfully applied to image classification [7]. In standard co-training, two classifiers are trained on two different views of the same data. In its deep learning method counterpart, since we do not necessarily have different data views, DCT exploits adversarial examples to encourage the view difference. The networks, thus, provide different and complementary information about the data, and do not collapse.

In this article, we report experiments comparing DCT and MT on three benchmark audio datasets: Google Speech Commands v2 (GSC), UrbanSound8K (UbS8k), and Environmental Sound Classification (ESC-10). We varied the fraction of labeled data from 10% to 50%, to get an idea of what is the best proportion of labeled data needed by both approaches. We will also compare the results to the fully-supervised learning setting, where 100% of the labeled data is used for learning.

[1] https://github.com/leocances/Deep-Co-Training.git

2. METHOD OVERVIEW

2.1. Mean Teacher

MT uses two neural networks: a "student" f and a "teacher" g, that share the same architecture. The weights ω of the student model are updated using the standard gradient descent algorithm, whereas the weights W of the teacher model are the Exponential Moving Average (EMA) of the student weights. The teacher weights are computed at every mini-batch iteration t, as the convex combination of its weights at t - 1 and the student weights, with a smoothing constant α:

$$W_t = \alpha \cdot W_{t\text{-}1} + (1 - \alpha) \cdot \omega_t \tag{1}$$

There are two loss functions applied either on the labeled or unlabeled data subsets. On the labeled data, the usual cross-entropy (CE) is used between the student model's predictions and the ground-truth. The labeled data are represented by x_s the ground truth by y_s:

$$\mathcal{L}_{\text{sup}} = \text{CE}\big(f(x_s), y_s\big) \tag{2}$$

The consistency cost is computed from the student prediction $f(x_u)$ and the teacher prediction $g(x'_u)$. We used a Mean Square Error (MSE), x_u is the non-labeled dataset, and x'_u the same set but slightly perturbed with Gaussian noise and a 15 dB signal-to-noise ratio.

$$\mathcal{L}_{\text{cc}} = \text{MSE}\big(f(x_u), g(x'_u)\big) \tag{3}$$

The final loss function is the sum of the supervised loss function and the consistency cost weighted by a factor λ_{cc} which controls its influence.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}} + \lambda_{\text{cc}} \cdot \mathcal{L}_{\text{cc}} \tag{4}$$

2.2. Deep Co-Training

DCT has been recently proposed by Qiao and colleagues [7]. It is based on Co-Training (CT), the well-known generic framework for SSL proposed by Blum and colleagues in 1998 [6]. The main idea of Co-Training is based on the assumption that two independent views on a training dataset are available to train two models separately. Ideally, the two views are conditionally independent given the class. The two models are then used to make predictions on the non-labeled data subset. The most confident predictions are selected and added to the labeled subset. This is an iterative process.

DCT is an adaptation of CT in the context of deep learning. Instead of relying on views of the data that are different, DCT makes use of adversarial examples to ensure the independence in the "view" presented to the models. The second difference is that the whole non-labeled dataset is used during training. Each batch is composed of a supervised and an unsupervised part. Thus, the non-labeled data are directly used, and the iterative aspect of the algorithm is removed.

Let $\mathcal{S}$ and $\mathcal{U}$ be the subsets of labeled and unlabeled data, respectively, and let f and g be the two neural networks that are expected to collaborate.

The DCT loss function is comprised of three terms, as shown in Eq. 5. These terms correspond to loss functions estimated either on $\mathcal{S}, \mathcal{U}$, or both. Note that during training, a mini-batch is comprised of labeled and unlabeled samples in a fixed proportion. Furthermore, in a given mini-batch, the labeled examples given to each of the two models are sampled differently.

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \lambda_{\text{cot}}\mathcal{L}_{\text{cot}} + \lambda_{\text{diff}}\mathcal{L}_{\text{diff}} \tag{5}$$

The first term, $\mathcal{L}_{\text{sup}}$, given in Eq. 6, corresponds to the standard supervised classification loss function for the two models f and g, estimated on examples x_1 and x_2 sampled from $\mathcal{S}$. In our case, we use categorical Cross-Entropy (CE), the standard loss function used in classification tasks with mutually exclusive classes.

$$\mathcal{L}_{\text{sup}} = \text{CE}\big(f(x_1), y_1\big) + \text{CE}\big(g(x_2), y_2\big) \tag{6}$$

In SSL and Co-Training, the two classifiers are expected to provide consistent and similar predictions on both the labeled and unlabeled data. To encourage this behavior, the Jensen-Shannon (JS) divergence between the two sets of predictions is minimized on examples x_u sampled from the unlabeled subset $\mathcal{U}$ only. Indeed, there is no need to minimize this divergence also on $\mathcal{S}$ since $\mathcal{L}_{\text{sup}}$ already encourages the two models to have similar predictions on $\mathcal{S}$. Eq. 7 gives the JS analytical expression, with H denoting entropy.

$$\begin{aligned}\mathcal{L}_{\text{cot}} \;=\; & H\Big(\frac{1}{2}\big(f(x_u) + g(x_u)\big)\Big) \\ & - \frac{1}{2}\Big(H\big(f(x_u)\big) + H\big(g(x_u)\big)\Big)\end{aligned} \tag{7}$$

For DCT to work, the two models need to be complementary: on a subset different from $S \cup U$, examples misclassified by one model should be correctly classified by the other model [8]. This can be achieved in deep learning by generating adversarial examples with one model and training the other model to be resistant to these adversarial samples. To do so, the $\mathcal{L}_{\text{diff}}$ term (Eq. 8) is the sum of the Cross-Entropy losses between the predictions $f(x_1)$ and $g(x'_1)$, where x_1 is sampled from $S \cup U$ and x'_1 is the adversarial example generated with model f and x_1 taken as input. The second term is the symmetric term for model g.

$$\mathcal{L}_{\text{diff}} \;= \text{CE}\big(f(x_1), g(x'_1)\big) + \text{CE}\big(g(x_2), f(x'_2)\big) \tag{8}$$

For the adversarial generation, we use the Fast Gradient Signed Method (FGSM, [9]), as in Qiao's work.

For more in-depth details on the technical aspects of DCT, the reader may refer to [7]. We implemented DCT as closed as described in Qiao's article, using PyTorch, and made sure to accurately reproduce their results on CIFAR-10: about 90% accuracy when using only 10% of the training data as labeled data (5000 images).

3. EXPERIMENTS

We carried out experiments on three datasets: Environmental Sound Classification dataset (ESC-10), and UrbanSound8K (UbS8K) and the Google Speech Commands v2 dataset (GSC). For each of them, different fractions of the whole labeled training subset were used to simulate different SSL settings. The labeled fraction files are randomly sampled while preserving the original class distributions. Each mini-batch contains both labeled and unlabeled samples in the same proportions as the whole training subsets. The 10% (25%, 50%) setting refers to a setting where only 10% (25%, 50%) of the labeled data is used. The supervised models are trained using that amount of data only. The MT and DCT models used the full 100% for training, with 10% (25%, 50%) being labeled and the rest unlabeled.

3.1. Datasets

Google Speech Commands Dataset v2 [10] is an audio dataset of spoken words designed to evaluate keyword spotting systems. The dataset is split into 85511 training files, 10102 validation files, and 4890 testing files. The latter is used for the evaluation of our systems. We ran the task of classifying the 35 word categories of this dataset. The files are zero-padded to 1 second if needed and sampled at 16 kHz before being converted into 32×64 log-Mel spectrogram.

UrbanSound8k [11] is a dataset composed of 8742 files between 1 and 4 seconds long separated into 10 evenly sized categories. The dataset is provided with ten cross-validation files of uniform size that will be used for system evaluation. The files are zero-padded to 4 seconds, resampled to 22 kHz, and converted to 431×64 log-Mel spectrograms.

Environmental Sound Classification 10 Dataset [12] is a selection of 400 5-second-long recordings of audio events separated into 10 evenly sized categories. The dataset is provided with five uniformly sized cross-validation folds that will be used to perform the evaluation. The files are sampled at 44 kHz and are converted into 431×64 log-Mel spectrograms.

The 64 Mel-coefficients were extracted using a window size of 2048 samples and a hop length of 512.

3.2. Experimental Setup

We use wideresnet28_2 [13] architecture in all our experiments. It is very efficient, achieving SOTA performance on the three datasets. Moreover, its small size allows to experiment quickly. It consists of three groups of four blocks of convolutions connected to a fully connected layer. Each convolution block is composed of a convolution followed by batch normalization, the ReLU activation, and max-pooling, applied after each block. It is comprised of about 1.4 Million parameters. We used the implementation available in PyTorch [14].

The models are trained using the ADAM optimizer. Table 1 gives the hyperparameter values, that were set with grid search performed on UbS8K. Those were also used on the two other datasets. Besides these values, the learning rates were weighted by a descending cosine rule defined such as $\text{lr} = 0.5(1.0 + \cos((T - 1) \times \pi/\text{nb\_epoch}))$. The loss terms in MT and DCT are weighted by the λ ratios, as given in Eqs. 4 and 5. These ratios ramp up to their maximum value within a warm-up length wl reported in Table 1. In MT, the maximum value for λ_{cc} is 1 and α is set to 0.999. For DCT, the maximum values for λ_{cot} and λ_{diff} are 1 and 0.5. The ramp-up is defined by $\lambda(\text{epoch}) = \lambda_{\max}\left(1 - e^{-5\times(1-(\text{epoch}/\text{wl}))}\right)$.

3.3. Evaluation

For Ubs8K and ESC-10, we used the official cross-validation folds. We report averaged classification accuracy with standard deviation. For GSC, we report accuracy averaged over five runs, we do not report standard deviation since it is almost always 0.1%. Overall, MT and DCT achieved greater accuracy than supervised training with DCT systematically outperforming MT.

The results for GSC are reported in Table 2. In the 10% setting, the supervised approach, MT and DCT reached 90.38%, 91.49%, 93.82%, respectively. These scores represent relative gains of 1.2% for MT and 3.8% for DCT. These relative gains decrease as the number of labeled files used increases. In the 50% setting, they reached 0.6% for MT and 0.8% for DCT.

UrbanSound8k, whose scores are presented in Table 3, is a smaller dataset that benefits more from the

Table 1. Training parameters used on the three datasets. Bs: batch size, lr: learning rate, wl: warm-up length in epoch, e: number of epochs.

	bs	lr	wl	e
Supervised	64	0.003	-	100
MT	64	0.001	50	200
DCT	100	0.0005	160	300

Table 2. Accuracy (%) on Google Speech Commands.

Labeled fraction	Supervised	MT	DCT
10%	90.38	91.49	93.82
25%	92.89	93.71	94.83
50%	94.17	94.72	94.89
100%	95.58	-	-

semi-supervised approaches. The DCT gains varied from 7.0% (10% setting) to 4.3% (50% setting), while for MT, the gains varied from 4.5% to 1.6%. We can also note that when in the 50% setting, DCT achieved better accuracy than the purely supervised model. This, to a lesser extent, can also be observed with MT.

Table 3. Accuracy (%) on UrbanSound8K.

Labeled fraction	Supervised	MT	DCT
10%	67.88 ± 4.31	67.75 ± 4.35	72.64 ± 5.24
25%	72.23 ± 4.75	75.45 ± 3.95	76.01 ± 5.40
50%	73.66 ± 5.26	74.87 ± 5.45	76.85 ± 4.85
100%	74.44 ± 4.88	-	-

The ESC-10 dataset, on which the results are reported in Table 4, has only 320 training files. In the 10% setting, each class is represented by three files only. It is also on this dataset that the most significant accuracy gains were observed. For MT, the relative gains varied from 7.6% to 1.6%, and for DCT, from 11.7% to 3.9%.

These results tend to show that DCT outperforms MT, at least on the three datasets used in this work. In [15], MT has been used on UrbanSound8K (Kaggle variant) and GSC (v1) without the same dataset configurations as we did here, so that the results are not directly comparable to ours. Nevertheless, the authors report further gains using audio augmentation in MT. We plan to explore this in future work.

Figure 1 shows the training times of MT and DCT on the two largest datasets, using an Nvidia Quadro RTX6000 GPU. Training on ESC-10 took less than 30 minutes for DCT (longest method); therefore, we did not display the training times for ESC-10. The semi-supervised methods' complexity implies longer training times, up to six times longer for DCT and five times longer for MT on GSC.

Table 4. Accuracy (%) on ESC-10.

Labeled fraction	Supervised	MT	DCT
10%	67.81 ± 4.04	72.97 ± 7.94	75.78 ± 5.30
25%	82.50 ± 5.73	85.16 ± 3.41	89.22 ± 3.96
50%	88.28 ± 2.92	89.69 ± 4.98	91.72 ± 5.14
100%	91.72 ± 1.96	-	-

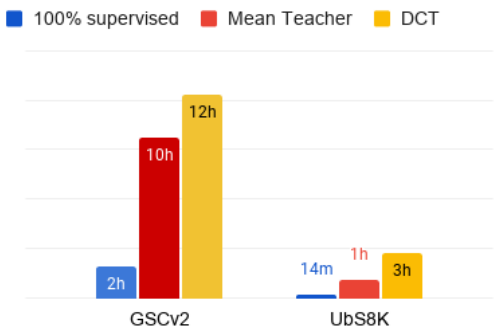

Fig. 1. Comparison of training times between supervised learning, MT and DCT (w/o cross-validation).

4. CONCLUSION

In this paper, we showed the effectiveness of Deep Co-Training applied to audio tagging. Promising results were obtained on three datasets of different sizes and containing sounds of very different sources, namely speech (Google Speech Command), urban noises (UrbanSound8k), and more general noises (ESC-10). DCT consistently outperformed Mean Teacher, an approach used recently by the AT community. For instance, using a fraction of 10% of labeled data, DCT yielded a maximum gain of 11.7% for ESC-10, 7.0% for UbS8K, and 3.8% for GSC. With 50% of labeled data, DCT even exceeded the performance of a purely supervised model.

A number of SSL methods for image classification task have been proposed recently, such as MixMatch [16] and FixMatch [17]. They both use of augmentations as their core mechanism. In [15], MT has been used in conjunction with augmentations that are reported to bring further improvements. We plan to add augmentations to DCT in short-term future work.

5. ACKNOWLEDGMENT

This work was partially supported by the Agence Nationale de la Recherche LUDAU (Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning) project (ANR-18-CE23-0005-01). We used the OSIRIM platform, administered by IRIT and supported by CNRS, the Region Midi-Pyrénées, the French Government, ERDF (http://osirim.irit.fr/site/en).

6. REFERENCES

[1] Z. Zhang and B. Schuller, "Semi-supervised learning helps in sound event classification," in *proc. ICASSP*, Kyoto, 2012, pp. 333–336.

[2] Wenjing Han, Eduardo Coutinho, Huabin Ruan, Haifeng Li, Björn Schuller, Xiaojie Yu, and Xuan Zhu, "Semi-supervised active learning for sound classification in hybrid learning environments," *PLOS ONE*, vol. 11, no. 9, pp. 1–23, 2016.

[3] D. Hakkani-Tur, G. Tur, M. Rahim, and G. Riccardi, "Unsupervised and active learning in automatic speech recognition for call classification," in *proc. ICASSP*, Montreal, 2004, pp. 429–432.

[4] Antti Tarvainen and Harri Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *proc. NeurIPS*, Long Beach, 2017, pp. 1195–1204.

[5] Lu JiaKai, "Mean teacher convolution system for dcase 2018 task 4," Tech. Rep., DCASE Challenge, Surrey, 2018.

[6] Avrim Blum and Tom Mitchell, "Combining labeled and unlabeled data with co-training," in *proc. COLT*, Madison, 1998, pp. 92–100.

[7] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille, "Deep co-training for semi-supervised image recognition," in *proc. ECCV*, Munich, 2018, pp. 135–152.

[8] Mark-A Krogel and Tobias Scheffer, "Multi-relational learning, text mining, and semi-supervised learning for functional genomics," in *Machine Learning*, 2004, pp. 61–81.

[9] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *proc. ICLR*, San Diego, 2015.

[10] Pete Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, arxiv:1804.03209.

[11] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello, "A dataset and taxonomy for urban sound research," in *proc. ACM Multimedia*, 2014, p. 1041–1044.

[12] Karol J. Piczak, "Esc: Dataset for environmental sound classification," in *proc. ACM Multimedia*, Brisbane, 2015, p. 1015–1018.

[13] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," in *proc. BMVC*, York, 2016, pp. 87.1–87.12.

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *proc. NeurIPS*, 2019, pp. 8026–8037.

[15] Kangkang Lu, Chuan-Sheng Foo, Kah Kuan Teh, Huy Dat Tran, and Vijay Ramaseshan Chandrasekhar, "Semi-supervised audio classification with consistency-based regularization.," in *proc. INTERSPEECH*, Graz, 2019, pp. 3654–3658.

[16] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *proc. NeurIPS*, Vancouver, 2019, pp. 5049–5059.

[17] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," 2020, arxiv: 2001.07685.