1 Supplementary Contents

1.1 Confusion of Terminology

The main motivation to introduce related concepts and settings (specifically, semi-supervised learning and meta-learning) is to connect them with low-shot learning. As an opinionated introduction to these terminologies, I believe that meta-learning can be thought of as a training strategy for low-shot learning, and semi-supervised learning corresponds exactly to the scenario that we face in the second stage of low-shot learning, as explained again below.

semi-supervised learning and low-shot learning. I said that the second stage in low-shot setting corresponds to semi-supervised learning: half-learn from labeled data and half-learn from unlabeled data. While this "half-half" explanation of semi-supervised learning is an informal and opinionated translation of "semi-supervised" to ease the understanding, I don't think what I said is incorrect. In the second stage in low-shot setting, labeled and unlabeled data are both available, and Hariharan and Girshick [1] do not specify how to train it: it is the freedom of algorithm implementors and network designers to choose how to do the training. It therefore, very naturally, can be thought of as a semi-supervised learning problem[1], where partially labeled data are available at training time. Kemker et al. also write that "Low-shot learning methods seek to accurately make inferences using a small quantity of annotated data. These methods typically build meaningful feature representations using unsupervised or semi-supervised learning to cope with the reduced amount of labeled data" [2].

few-shot and low-shot learning. Xuming said that few-shot and low-shot learning are of slight difference. This is what I feel too but I just can not find a clear distinction. However, Wang et al. said that "This challenge of learning new concepts from very few labeled examples, often called low-shot or few-shot learning, is the focus of this work" [4], which leads to a dangerous misunderstanding of low-shot and few shot learning being the same thing.

meta-learning and low-shot learning. The slide of introducing meta-learning is in Figure 1. Feeling sorry that I did not respect the standard definition, the intention of using "TRAIN" instead of "meta-train" as an opinionated introduction is as follows. By using "TRAIN",

[1] http://pages.cs.wisc.edu/~jerryzhu/pub/sslicml07.pdf

- it can be compared to the item "Transfer Learning (naive)" in the slide (it is not appropriate to use "meta-train" for transfer learning), and thus "TRAIN" can be easily understood as the standard training process.
- Therefore, there is no one asking, "what do you mean by 'meta-learn'?", a question that appeared in previous reading group when introducing meta-learning, if you recall.
- There is no need to follow the convention in which "task" is defined, which is a simplification for fresh students.

However, it turns out that I speak so fast that fresh students can not follow, and that people who already know meta-learning think it is weird. It is neither weird nor difficult. It is just different.

For those who want to read some papers where terminologies are misused as more counterexamples, have a look at Lipton and Steinhardt [3].

1.2 Further Explanation

- I made a typo in the slide as shown in Figure 2. The word "meta-learning" in the second item should be "TRAIN" or "meta-train".
- There is no experiment about the choice of cosine distance in [1], but there is one paragraph in [1] shown in Figure 3 concludes that network-based data augmentation might be better than common ones.
- the process of derivation of SGM loss in Figure 4 is just a mathematical formulation that gives us a loss function. We will calculate L_D^{SGM} for every tiny training subset S, and similarly in training we have to update the network parameters for each minibatch. Hence the final loss is a combination of the classification loss and L_D^{SGM}. You can check the well-structured and easy-to-follow source code[2],starting from **main.py** and try to locate the loss. What the code do is just what I said above.

1.3 A Little Disclaimer

The presenter is responsible for explaining clearly things he or she presents, but not for giving a story or explanation of "why it is as it is" if the paper author does not explain it. Realizing the differences between paper author and presenter, I suggest that

[2] https://github.com/facebookresearch/low-shot-shrink-hallucinate

Learning from data with annotations

Related Settings.

Let $\mathcal{Q} = \{S_i, T_i\}_{i=1,\ldots,n+1}$ be a list of training-test set pairs.

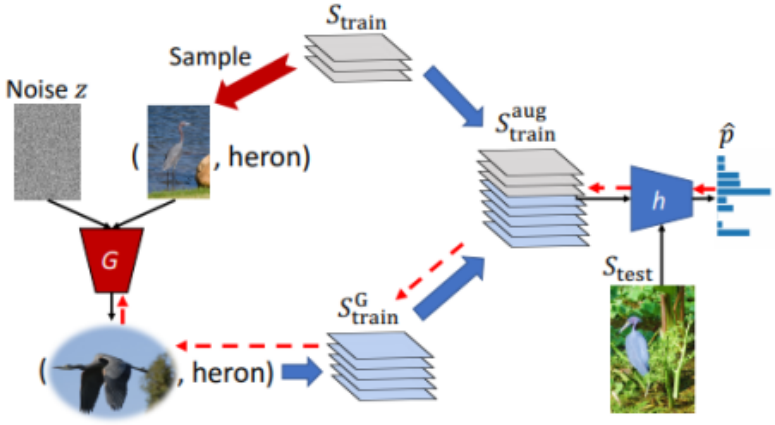
- Transfer Learning (naive)
  - TRAIN: train on S_1 and fine-tune on S_2.
  - TEST: test on T_2.
  - EXAMPLE: train on ImageNet, fine-tune and test on Cifar-10.
    - final layers replaced by a new classifier for new classes.
- Meta-Learning (informal)
  - TRAIN: train on S_i and test on T_i $(i = 1, 2, \ldots, n)$.
    - S_i, T_i randomly sampled from $\mathcal{Q}$.
    - test accuracy as a loss provides error signal via BP.
  - TEST: train on S_{n+1} and test on T_{n+1}.
  - REMARK: meta-learning as a low-shot training strategy
    - TRAIN $\Longleftrightarrow$ the first stage in low-shot setting
    - TEST $\Longleftrightarrow$ the second stage in low-shot setting

Figure 1: An informal introduction to meta-learning (I change the original word "simplified" to "informal"). It is called informal or simplified because I do not know what classes (novel) should be in S_{n+1} and T_{n+1} as a novel task.

- when referring to the slide itself, a question like "why do you make slides like this?" is ok.
- when referring to technical details, a question like "why the author ....." is preferred.
- a question without subject is perfect, e.g., "Does this method make sense?", "What is the use of this component for the entire system?"

Pseudo-data generation for low-shot learning

Wang et al., CVPR' 2018



1. noise z + labeled image from $S_{\text{train}} \overset{G}{\Rightarrow}$ pseudo-labeled image.
   - G is learned
2. meta-learning with augmented dataset $S^{\text{aug}}_{\text{train}}$ and S_{test}.

Figure 2: Pseudo-data generation process for low-shot learning

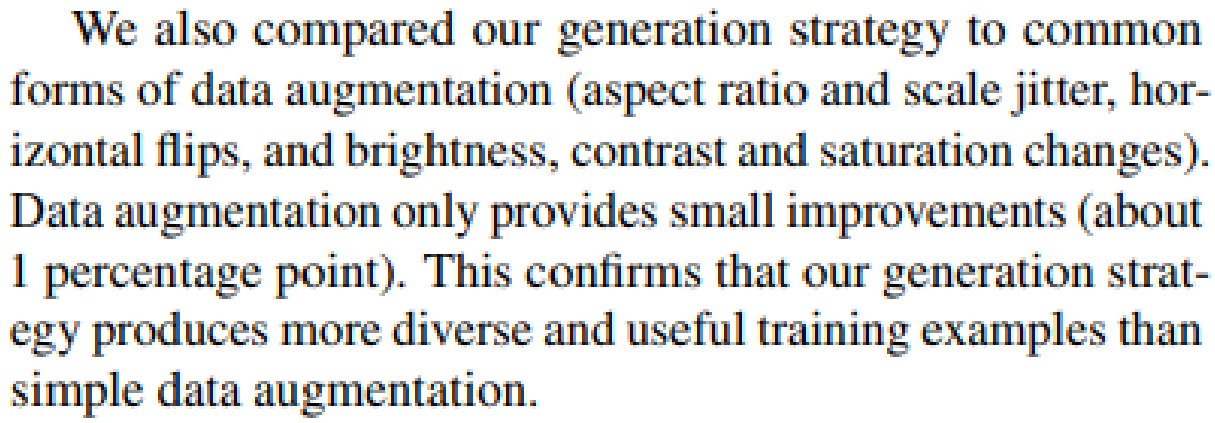


Figure 3: Pseudo-data generation process for low-shot learning

References

[1] B. Hariharan and R. B. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, pages 3037–3046, 2017.

[2] R. Kemker, R. Luu, and C. Kanan. Low-shot learning for the semantic segmentation of remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[3] Z. C. Lipton and J. Steinhardt. Troubling trends in machine learning scholarship. *arXiv preprint arXiv:1807.03341*, 2018.

[4] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. *arXiv preprint arXiv:1801.05401*, 2018.

Squared Gradient Magnitude Loss for Low-shot Learning

Hariharan and Girshick, ICCV' 2017

simulate low-shot learning experiments on the base classes:

- train a feature extractor ϕ and a classifier W on dataset D (i.e., labeled base classes).
  - $L_D(\phi, W) = \frac{1}{|D|}\Sigma_{(x,y)\in D} L_{cls}(W, x', y)$, where $x' = \phi(x)$.

- Let $S \subset D$ be a tiny training set.
  - loss for training a classifier V on S:

$$L_S(\phi, V) = \frac{1}{|S|}\Sigma_{(x,y)\in S} L_{cls}(V, \phi(x), y).$$

  - want the minimizer of $L_S(\phi, V)$ to match W, i.e., (note that $L_S(\phi, V)$ is convex in V):

$$\nabla_V L_S(\phi, V)|_{V=W} = 0 \Rightarrow L_D^{SGM} = \frac{1}{|D|}\Sigma_{(x,y)\in D} ||\nabla_V L_S(\phi, V)|_{V=W}||_2^2.$$

Figure 4: SGM loss for low-shot learning