SYNTHETIC LABELING: A NOVEL APPROACH TO ADVANCING FEW-SHOT LEARNING

Zhaoyan Lyu
University College London
z.lyu.17@ucl.ac.uk

Gholamali Aminian
The Alan Turing Institute
gaminian@turing.ac.uk

Miguel R. D. Rodrigues
University College London
m.rodrigues@ucl.ac.uk

ABSTRACT

In the field of few-shot learning, the scarcity of labeled data significantly hinders progress. This paper introduces an innovative regularization algorithm designed to enhance generalization performance in classification tasks by leveraging synthetically labeled data. The approach utilizes a single encoder and multiple decoders, trained on both an original dataset with ground truth labels and synthetic datasets with artificial labels. Our empirical studies demonstrate that this method effectively improves neural network generalization, both independently and when integrated with other regularizers. This versatility underscores the potential of synthetic labeling in overcoming data limitations in few-shot learning scenarios.

1 INTRODUCTION AND RELATED WORK

Learning under few data samples is known as *Few-shot learning*. There are various well-known approaches to learning with small (labeled) datasets. Data augmentation Shorten & Khoshgoftaar (2019) algorithms augment the training dataset, which contains data samples labeled accurately. In particular, they either manually filter, transpose, flip, rotate, erase, crop or color-shift the input image or use some deep learning approaches to transform the input image into a related domain. However, the ground-truth labeling is rarely augmented or tuned in such augmentation methodologies. In transfer learning Gupta et al. (2020) and meta-learning Hochreiter et al. (2001), one could train a network with several datasets generated from different distributions, which can also be seen as an augmentation of the dataset. Meanwhile, the weakly supervised learning Zhou (2018) and semi-supervised learning Zhu (2005) algorithms are used to handle the unlabeled or corrupt-labeled datasets. However, all the algorithms mentioned above are based on a dataset with ground-truth labeling. More related works are provided in Appendix A.

Unlike most previous papers on few-shot learning that worked with a true label dataset or a dataset with shifted distribution, we focus on approaches to learn models leveraging both the original dataset, containing ground truth labels, and synthetic datasets, containing synthetically created labels. When utilizing a dataset with synthetic labels as a regularizer, the network will be able to learn the patterns and features shared by the ground-truth labeled dataset and synthetically labeled dataset.

2 PROPOSED APPROACH

As shown in Figure 1, the proposed approach *trains based on the true and synthetic labels (TTSL)* involves three blocks: an *encoder* (blue) that encodes the input image to a representation, a *true decoder* (red) which predicts the ground-truth labels, and a *synthetic decoder* (pink) that predicts the synthetic labels. The encoder and the decoders are parameterized by W_E, W_D^t, and W_D^s, respectively. During the training phase, all blocks will be optimized jointly. In particular, the parameters of the encoder and both true and synthetic decoders are optimized in order to minimize the convex combination of empirical risks with respect to true and synthetic labels. During the testing phase, only the encoder and the true decoder are required, ensuring that there is no additional computational overhead in this stage.

The input data are represented by a random variable $X \in \mathcal{X}$, where $\mathcal{X}$ denotes the input space. The true and synthetic labels are modeled by random variables $Y^t, Y^s \in \mathcal{Y}$, with $\mathcal{Y}$ being the output space. The true label predicted by the true decoder is denoted as $\hat{Y}^t$, and the synthetic label predicted

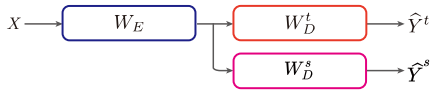

Figure 1: TTSL Algorithm Model

by the synthetic decoder as $\hat{Y}^s$. We define the true input-output pair as $Z = (X, Y^t) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The training set $S^t = \{Z_i^t = (X_i, Y_i^t)\}_{i=1}^n$ consists of true input-output data points sampled i.i.d. from $\mathcal{Z}$ according to distribution μ. The synthetic dataset $S^s = \{Z_i^s = (X_i, Y_i^s)\}_{i=1}^m$ shares inputs with the true dataset but has randomly generated labels $Y_i^s \in \mathcal{Y}$ that differ from Y_i^t, generated according to a separate distribution μ_s. In the experiments presented in the following section, μ_s is a uniform distribution.

The encoder and decoders are optimized to minimize the combination of the loss on each decoder defined as follows (where $(L(\cdot)$ represents cross-entropy loss):

$$\{W_E, W_D^t, W_D^s\} = \underset{W_E, W_D^t, W_D^s}{\arg\min} \frac{1}{n}\frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{m}(1-\beta)L(\widehat{Y}_i^t; Y_i^t) + \beta L(\widehat{Y}_j^s; Y_j^s) \tag{1}$$

3 Experiments and Discussion

We employed a subset of the Fashion-MNIST dataset Xiao et al. (2017) to test our proposed method. For this purpose, we randomly chose 1,000 images from the training set, which were then trained using a MLP-based neural network. Detailed procedures and configurations of our training methodology are comprehensively outlined in the appendix of our paper. Additionally, the appendix presents results from our experiments on the CIFAR-10 dataset, trained using the VGG-16 network, further demonstrating the versatility of our approach.

Table 1: MLP on subset of Fashion-MNIST

	Vanilla	Noisy∇	WD	dropout	DA	Mixup	LS
test loss	0.600±0.0015	0.596±0.0036	0.597±0.0013	0.605±0.0063	0.647±0.0052	0.834±0.0039	0.603±0.0029
GE	0.593±0.0008	0.565±0.0009	0.397±0.0008	0.591±0.0013	0.446±0.0013	0.503±0.0010	0.580±0.0008
+TTSL test loss	0.588±0.0026	-	0.585±0.0020	0.601±0.0060	0.645±0.0030	0.811±0.0025	0.580±0.0040
+TTSL GE	0.504±0.0010	-	0.260±0.0010	0.397±0.0010	0.150±0.0009	0.457±0.0003	0.509±0.0010
test acc.	0.786±0.0011	0.800±0.0020	0.799±0.0014	0.811±0.0043	0.797±0.0044	0.786±0.0039	0.810±0.0020
GE	0.210±0.0004	0.180±0.0010	0.164±0.0010	0.189±0.0011	0.149±0.0008	0.210±0.0005	0.190±0.0011
+TTSL test acc.	0.812±0.0023	-	0.801±0.0021	0.820±0.0031	0.800±0.0035	0.796±0.0021	0.811±0.0020
+TTSL GE	0.173±0.0006	-	0.110±0.0011	0.120±0.0022	0.091±0.0010	0.178±0.0003	0.189±0.0009

Table 1 demonstrates that our proposed algorithm not only get lower test loss and smaller generalization error (GE) in both test scenarios but also integrates effectively with other regularizers, including adding Gaussian noise to gradient (Noisy∇), weight decay (WD), dropout, data augmentation (DA), Mixup Carratino et al. (2020), and label-smoothing (LS) Szegedy et al. (2016). The hyper-parameters for each regularizer were selected through a grid-search process.

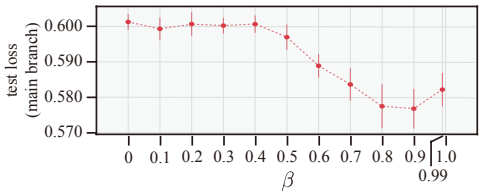

Figure 2: MLP trained on subset of Fashion-MNIST with various β.

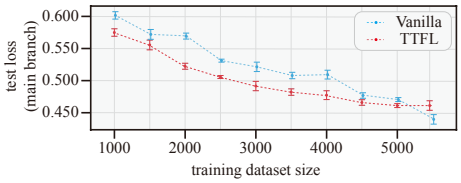

Figure 3: MLP trained on subset of Fashion-MNIST with various training dataset size.

Our ablation study shown in Figure 2 and Figure 3, which varied both β and the training dataset size, revealed that the proposed method surpasses vanilla training (when $\beta = 0$), especially in contexts with smaller training datasets. Additional ablation studies are presented in the Appendix C to further confirm the effectiveness of the proposed algorithm.

4 CONCLUSION

We introduced a method that employs synthetic labels and a branched neural network as a regularization strategy to enhance the generalization performance of neural networks in data-constrained regimes. Our experiments demonstrate the method's adaptability and compatibility with established regularization techniques, such as weight decay and dropout, highlighting its potential for real-world applications.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Nihar Bendre, Hugo Terashima Marín, and Peyman Najafirad. Learning from few samples: A survey. *arXiv preprint arXiv:2007.15484*, 2020.

Luigi Carratino, Moustapha Ciss'e, Rodolphe Jenatton, and Jean-Philippe Vert. On mixup regularization. *J. Mach. Learn. Res.*, 23:325:1–325:31, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 289–293. IEEE, 2018.

Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

Aakriti Gupta, Kapil Thadani, and Neil O'Hare. Effective few-shot classification with transfer learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1061–1066, 2020.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in Neural Information Processing Systems*, 31:10456–10465, 2018.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pp. 87–94. Springer, 2001.

Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11719–11727, 2019.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. Data augmentation using gans. *arXiv preprint arXiv:1904.09135*, 2019.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13001–13008, 2020.

Xueyuan Zhou and Mikhail Belkin. Semi-supervised learning. In *Academic Press Library in Signal Processing*, volume 1, pp. 1239–1269. Elsevier, 2014.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1): 44–53, 2018.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

Xiaojin Jerry Zhu. Semi-supervised learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences*, 2005.

A RELATED WORKS

Data Augmentation: When we do not have access to sufficient data, data augmentation is a common practice. There are different data augmentation techniques, depending on the data domain. For example, for imagery data, image rotation, clipping, and other techniques can be applied to create an augmented dataset (Shorten & Khoshgoftaar, 2019; Perez & Wang, 2017). Random erasing is also an effective method for image data augmentation (Zhong et al., 2020). For text classification tasks, multiple techniques, including synonym replacement, random insertion, random swap, and random deletion can also be used for data augmentation (Wei & Zou, 2019). Generative adversarial networks can also be applied to generate new data samples from the same domain to help to augment the dataset (Frid-Adar et al., 2018; Tanaka & Aranha, 2019). As listed above, the data augmentation techniques usually only transform the input data, leaving the original labeling untouched. However, our algorithm proposes to use fake labels for training.

Meta and Transfer Learning: Some works (Bendre et al., 2020; Wang et al., 2020) use transfer learning or meta-learning techniques to solve few-shot learning problems. The core idea of transfer learning and meta-learning in solving the problem of insufficient data is borrowing the knowledge learned from one or multiple other similar dataset sources. For example, Model-Agnostic Meta Learning (Finn et al., 2017) uses a gradient-based approach to learn from multiple tasks. Task-Agnostic Meta learning (Jamal & Qi, 2019) uses an entropy-based approach to few-shot learning. However, these approaches require multiple well-labeled datasets akin (i.e. with a similar data-generating distribution) to the target dataset, which is not always available. Our approach only uses the available dataset.

Semi and Weakly Supervised Learning: Semi-supervised and weakly supervised learning can also be related to our work because they use a dataset that is partly labeled or cheaply labeled. In particular, Semi-supervised learning (Zhu & Goldberg, 2009; Zhou & Belkin, 2014) leverages unlabeled data by using some techniques, e.g., entropy minimization (Grandvalet et al., 2005) and Pseudo-labeling (Lee et al., 2013). On the other hand, Zhou (2018); Hendrycks et al. (2018) deal with weakly-supervised learning, whose labels are cheaply labeled or contain considerable noise. However, in our setup, our (typically small) dataset is fully well-labeled, and we use a fake labeling process to achieve better performance.

B TRAINING DETAILS

B.1 TRAINING MLP WITH FASHION-MNIST DATASET

In the training of the MLP model on the Fashion-MNIST dataset, we employed a specific network architecture, which is demonstrated in Figure 4. The training dataset comprised 1,000 randomly selected samples. We used the Adam optimizer with a learning rate of 0.0001, running for 200 epochs.

B.2 TRAINING VGG-16 WITH CIFAR-10 DATASET

For the VGG-16 network trained on the CIFAR-10 dataset (which will be used in the next section), we similarly used a subset of 1,000 randomly chosen images from the training dataset. The network was optimized using the Adam optimizer, with a learning rate initially set to 0.001 for a total of 150 epochs. The architecture of the VGG-16 are shown in the Figure 5. In particular, there are various locations where the synthetic decoder can be branched, which will be explored in the ablation study presented in the following section.

C MORE ABLATION STUDY

C.1 VGG-16 BRANCHED AT VARIOUS LOCATIONS

We evaluated the impact of branching the synthetic decoders at different layers within the VGG-16 network on performance. The outcomes are detailed in Table 2. The results indicate that our proposed method enhances classification accuracy regardless of the layer at which the true branch

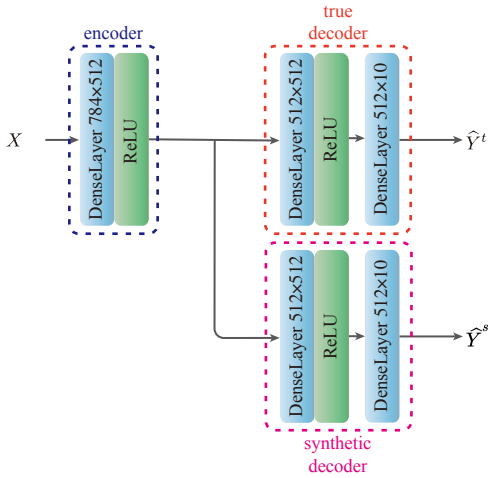

Figure 4: Architecture of the MLP trained for Fashion-MNIST classification task.

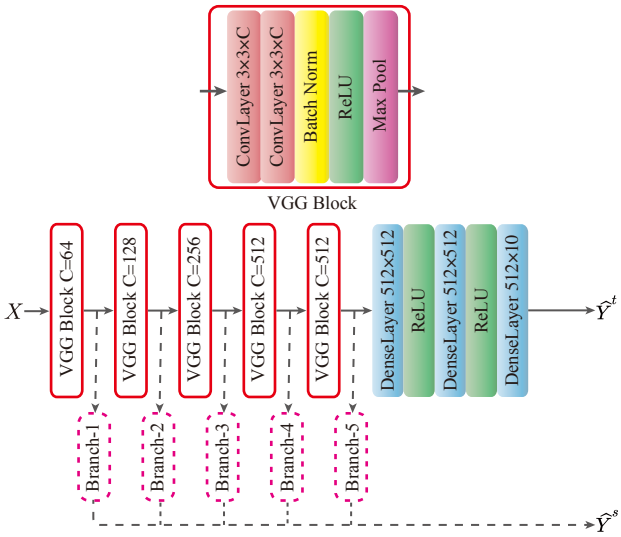

Figure 5: Architecture of the VGG-16 trained for CIFAR-10 classification task.

is implemented. However, the extent of improvement varies depending on the specific configuration of the setup.

C.1.1 MULTIPLE SYNTHETIC DECODERS

Given that synthetic labels are assigned randomly, it is feasible to deploy multiple synthetic decoders at the same layer. As illustrated in Figure 6, our experiment with an MLP trained on MNIST classification demonstrates that a model with two synthetic branches yields the best performance. However, it is noteworthy that all configurations surpass the baseline vanilla setup in terms of performance. It should be noted, though, that employing a synthetic decoder introduces higher performance variance, which is sensitive to the initialization.

Table 2: CIFAR-10 classification results: The three major columns are the TTSL algorithm compared with vanilla dataset, the TTSL algorithm combined with weight decay regularization and dropout. The first row of each major column is the results of regularization alone (or no regularization), without applying the TTSL algorithm. The other rows are the results of choosing different location to branch out from the main VGG network as shown in 5

TTSL			TTSL + WD			TTSL + dropout		
	acc	GE		acc	GE		acc	GE
Vanilla	0.3450	0.6550	WD	0.3494	0.6496	dropout	0.3636	0.6234
Branch-1	0.3932	0.6068	Branch-1	**0.3788**	0.5972	Branch-1	0.3700	0.6260
Branch-2	0.3658	0.6342	Branch-2	0.3534	0.6236	Branch-2	0.3772	0.6038
Branch-3	0.3882	0.6118	Branch-3	0.3650	0.6100	Branch-3	0.3810	**0.5900**
Branch-4	**0.3990**	**0.6010**	Branch-4	0.3606	0.6114	Branch-4	0.3850	0.6110
Branch-5	0.3514	0.6255	Branch-5	0.3696	**0.5704**	Branch-5	**0.3894**	0.6106

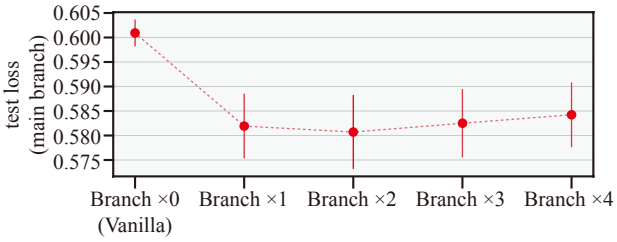

Figure 6: Fashion-MNIST test loss with multiple synthetic decoders. We duplicate the synthetic decoders multiple times to create extra synthetic labeling. The synthetic decoders are all branched from the same layer as illustrated in 4. When we do not have any synthetic branches (Branch $\times 0$), the experiment is equivalent to vanilla training. The marker is the averaged value over 20 repeated experiments and the error bars represent the standard deviation.