Transfer of Pretrained Model Weights Substantially Improves Semi-Supervised Image Classification

Attaullah Sahito, Eibe Frank, and Bernhard Pfahringer

Department of Computer Science, University of Waikato, Hamilton, New Zealand.
a19@students.waikato.ac.nz, {eibe,bernhard}@waikato.ac.nz

Abstract. Deep neural networks produce state-of-the-art results when trained on a large number of labeled examples but tend to overfit when small amounts of labeled examples are used for training. Creating a large number of labeled examples requires considerable resources, time, and effort. If labeling new data is not feasible, so-called semi-supervised learning can achieve better generalisation than purely supervised learning by employing unlabeled instances as well as labeled ones. The work presented in this paper is motivated by the observation that transfer learning provides the opportunity to potentially further improve performance by exploiting models pretrained on a similar domain. More specifically, we explore the use of transfer learning when performing semi-supervised learning using self-learning. The main contribution is an empirical evaluation of transfer learning using different combinations of similarity metric learning methods and label propagation algorithms in semi-supervised learning. We find that transfer learning always substantially improves the model's accuracy when few labeled examples are available, regardless of the type of loss used for training the neural network. This finding is obtained by performing extensive experiments on the SVHN, CIFAR10, and Plant Village image classification datasets and applying pretrained weights from Imagenet for transfer learning.

Keywords: Semi-supervised learning, Transfer learning, Self-learning, Triplet loss, Contrastive loss, Arcface loss.

1 Introduction

Neural networks are frequently used for image classification tasks and yield state-of-the-art results in this application. However, for training, these models generally need a lot of labeled samples, and they tend to overfit on small amounts of labeled data. This problem is of particular importance when limited labeled samples are available due to time or financial constraints. Addressing this problem requires machine learning methods that are able to work with a limited amount of labeled data and also make efficient use of the side information available from unlabeled data.

Semi-supervised learning (SSL) aims to improve performance by exploiting both labeled and unlabeled examples. Given an input space X containing

the examples, SSL methods are designed to work with labeled examples $L = \{(x_1, y_1), (x_2, y_2), ..., (x_{|L|}, y_{|L|})\}$ and unlabeled examples $U = \{x_1^{'}, x_2^{'}, ..., x_{|U|}^{'}\}$, where $x_i, x_j^{'} \in X$ with $i = 1, 2, ..., |L|$ and $j = 1, 2, ..., |U|$ and y_i are the labels of x_i, with $y_i \in \{1, 2, 3, ..., c\}$ (c being the number of classes).

A few assumptions are required to make semi-supervised learning a principled approach [3]:

1. If two instances x_1, x_2 are close in a high-density region, then their corresponding outputs y_1, y_2 should also be close.
2. If instances are in the same structure (referred to as a cluster or manifold), they are likely to be of the same class.
3. The decision boundary between classes should lie in a low-density region of the input space.

Almost all standard neural networks for image classification are trained by minimising cross-entropy loss on labeled training data. In this paper, along with cross-entropy loss, we also consider another class of losses, comprising so-called similarity metric learning losses, which operate on the relationships between samples such that instances of the same class are considered similar and those belonging to different classes are considered dissimilar. Once a similarity function has been trained, which is parameterised by a neural network, feature vectors (embeddings) of examples produced by the network will be grouped together according to class labels, normally in Euclidean space. These learned embeddings lend themselves naturally to semi-supervised learning because they can be employed to assign class labels to unlabeled examples using very simple classification methods such as nearest-neighbor classifiers.

This approach is related to work on pseudo-labeling [10,16], where the model is initially trained on limited data. However, in this paper, instead of applying random initialisation of network parameters when training starts, we investigate using pretrained weights from another domain and show that this provides much better generalisation ability. Using pretrained model weights is a standard approach for transfer learning in supervised settings, but appears to have received little attention in the context of semi-supervised learning, particularly when applying self-learning with metric learning.

We use a pretrained neural network model trained on Imagenet [17]. A schematic overview of the proposed approach is shown in Figure 1. Fine-tuning on data from the target domain is performed on the (very small) initial set of labeled examples. Following that, confident predictions for unlabeled examples are added to labeled examples for iterative retraining of the neural network—this is the standard self-learning method for semi-supervised learning. It enables us to obtain more labeled training data and the assumption is that this eventually helps in achieving significant performance improvements. In our experiments on image classification tasks, we compare using pretrained weights for the neural network to random initialisation of the weights.

The main contribution of this work is an extensive empirical investigation of transfer learning in the context of self-learning. Using cross-entropy loss as well

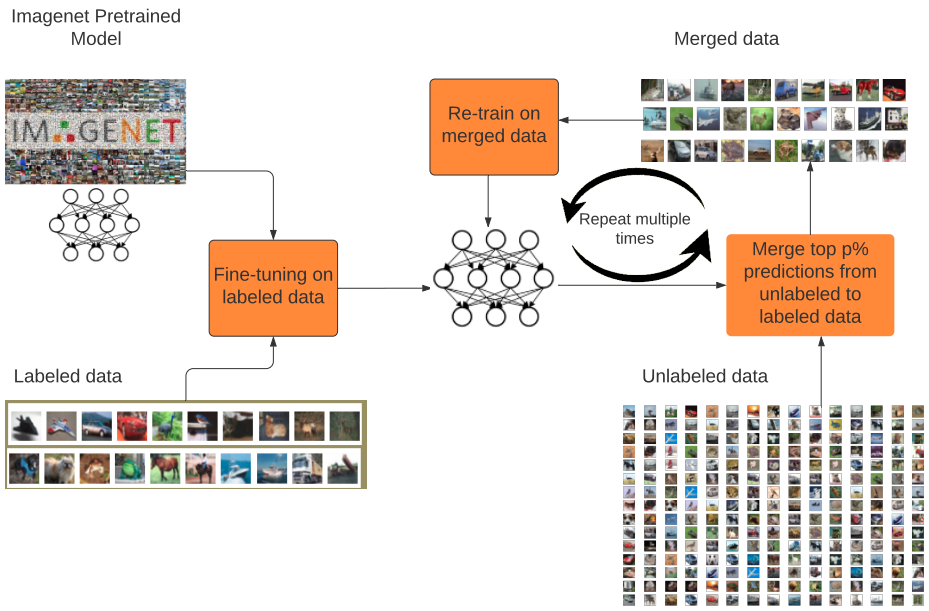


Fig. 1: Overview of the approach.

as combinations of similarity metric learning losses (e.g., triplet loss, contrastive loss, and Arcface loss) with simple nearest-neighbor-based label propagation, we find that transfer learning always substantially improves the classification accuracy of the model when few labeled examples are available, regardless of which loss function is used for training the neural network. More specifically, for semi-supervised learning using self-learning on the SVHN, CIFAR10, and Plant Village image classification datasets, we obtain a substantial improvement using pretrained weights when few labeled examples are available for training. Thus, our results indicate that the well-established method of performing transfer learning by re-using pretrained weights—commonly applied when performing a purely supervised training of a neural network—is particularly useful in the context of semi-supervised learning.

2 Related Work

In this section, we briefly discuss some existing work on semi-supervised learning and transfer learning.

2.1 Semi-supervised Learning

Semi-supervised Learning (SSL) lies between supervised and unsupervised learning. SSL tries to employ labeled examples as well as unlabeled examples for more accurate prediction. There are many different techniques available from the literature on SSL using deep neural networks. Some employ autoencoders [12,15], others use generative models [5,23,20] or are based on regularization ideas [14,19]. In pseudo-labeling [10], the model is trained on the limited labeled data first and then re-trained on an extended set of labeled data, based on the predictions of the original model for the unlabelled training data.

Our method builds on work investigating transferring learning using both cross-entropy loss and similarity-based metric learning with neural networks. Pair and triplet based loss functions provide the foundation for standard approaches to metric learning. A classic pair-based method is to use contrastive loss [4], which tries to bring similar pairs closer and push farther away dissimilar pairs. Pairs can be extended to triplets. They consist of an anchor, a positive, and a negative example, where the anchor is more similar to the positive example than the negative one. The resulting triplet loss function [21] was originally used on triplets of images for face verification. Metric learning-based loss functions [24] have also been successfully employed for image classification.

Another related class of metric learning methods are based on modified classification losses. Examples include Arcface [6], Sphereface [11] and Cosface [22]. For metric learning, Arcface, Sphereface, and Cosface apply multiplicative-angular, additive-cosine, and additive-angular margins, respectively.

2.2 Transfer Learning

Since the successful Imagenet challenge [17], transfer learning has been used widely in visual recognition tasks such as object detection [7]. Transfer learning uses the network weights learned by training on the large and labeled Imagenet dataset and fine-tunes the weights for the respective target domain. When the target domain is sufficiently closely related to the source domain of Imagenet, then transfer learning usually generalizes much better than training from scratch on the smaller target domain alone.

3 Semi-supervised Learning using Self-learning

The semi-supervised learning approach we apply is based on self-learning. The model is initially trained using a limited number of labeled examples. Then confident predictions for unlabelled examples are added to the set of labeled examples for retraining of the model. Generally, multiple iterations of labelling and retraining are performed. One important hyper-parameter is the selection percentage p, which specifies how many of the most confident predictions are added to the training set after each iteration. We use a small value of p in our experiments to select the most confident predictions only. Generating many

more labeled data points in this fashion allows for deep neural networks to be trained to their full capacity, and generally results in significant performance improvements. For more details on this approach see, for instance, our previous work in [18].

In this paper, for network weight initialisation, we transfer pretrained weights from Imagenet classification and fine-tune on the target domain. We compare the performance achieved by this weight transfer to the performance of training using a fully random initialisation of the weights of the neural network.

The proposed approach is very general, suggesting that a spectrum of loss functions and label propagation algorithms can all work well in this framework. We use the most widely used classification loss, i.e., softmax cross-entropy, as a first option. In addition, we explore loss functions based on similarity metric learning. The embeddings produced by the neural network after training with a similarity function can be employed to assign class labels to unlabeled examples using very simple classification methods such as a nearest-neighbor classifier. Below we review the loss functions used for the experiments.

3.1 Softmax Cross-entropy Loss

The single most frequently used classification loss function is softmax cross-entropy, which is a measure of the difference between the desired probability distribution and the predicted probability distribution:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{c} e^{W_j^T x_i + b_j}}, \tag{1}$$

where $x_i \in \mathbb{R}^d$ denotes the deep features (the "embedding") of the i^{th} sample, belonging to the class y_i, and d is the dimension of the embedding, $W_j \in \mathbb{R}^d$ denotes the j^{th} column of the weight matrix $W \in \mathbb{R}^{d \times c}$ and $b_j \in \mathbb{R}^c$ is the bias term. The batch size for gradient descent is N and c is the number of classes.

3.2 Siamese Networks

Siamese networks [2] are neural networks for training a similarity function given labeled data using one of several possible loss functions. They can be thought of as two identical copies of the same network, sharing all weights. They are particularly suitable for datasets with many classes containing only a few labeled instances per class and can employ any of the loss functions listed below.

Triplet Loss The triplet loss [21] is widely used. A triplet's anchor example a, positive example p, and negative example n are provided as a training example to the network for getting corresponding embeddings. Normally a and p come from the same class, and n is from a different class. Triplet loss tries to push the negative example's embedding farther away from positive example's one, with a

user-specified minimum margin m. Using, e.g., Euclidean distance $d(.,.)$ between embedded examples, the triplet loss is calculated as:

$$\mathcal{L} = max(d(a,p) - d(a,n) + m, 0) \tag{1}$$

Triplet loss tries to push $d(a,p)$ to 0 and $d(a,n)$ to be greater than $d(a,p)+m$. Triplets can be categorized as:

- **Easy triplets**: those with a loss of 0.
- **Hard triplets**: those where n is closer to a than p.
- **Semi-hard triplets**: those where n is not closer to a than p, but is within the margin, thus still returning a positive loss.

In our experiments, we use semi-hard triplets for training of the neural network as they yield more distinctive embeddings [21].

Contrastive loss The contrastive loss [8] is a pair-based loss that attempts to bring similar examples closer to each other and push dissimilar examples farther away with respect to a minimum margin m. Contrastive loss for embeddings of two examples x_1 and x_2 can be calculated as follows:

$$\mathcal{L} = y \times d(x_1, x_2) + (1 - y) \times max(0, m - d(x_1, x_2)) \tag{2}$$

Here, $y = 1$ if x_1 and x_2 are from the same class, and $y = 0$ otherwise.

ArcFace loss Arcface loss [6] is a modified cross-entropy loss with angular margins in the softmax expression, which is designed for improved discrimination in metric learning. The loss is calculated as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{c} e^{s\cos\theta_j}}. \tag{3}$$

θ_j is the angle between the l_2-normalized weight vector W_j and the feature vector x_i. The bias term b_j is ignored for simplicity. The feature vector x_i is l_2-normalised and scaled to s, the radius of the hypersphere. An additive angular margin penalty m is added to the ground truth angle θ_{y_i}.

4 Experiments

For evaluating the effect of transfer learning, we consider three image classification problems. For all datasets, a small subset of labeled examples was chosen according to standard semi-supervised learning practice, with a balanced number of examples from each class. All remaining examples were used as unlabeled training examples. For triplet, contrastive and Arcface loss, k-nearest neighbor is used for label prediction, with $k = 1$ for simplicity. We always include two network version in the comparison: one using randomly initialised weights, and

one using pretrained weights from ImageNet. All models are evaluated on the standard test split for each dataset in three different ways: after training only on the initially labeled examples, then after training for a number of meta-iterations using our semi-supervised learning approaches, and also — for comparison — after training on all labeled training examples. The two sets of results computed from a) only the initial labeled examples, and b) all labeled training examples, act as an empirical lower and upper bound for the semi-supervised approaches.

We used the VGG16 network architecture for all experiments. A fully connected layer is added at the end of the model for generating a 256-dimensional embedding space. A mini-batch size of 100 is used for all the experiments. For updating the network parameters, Adam is used as the optimizer, except for contrastive loss, which uses Rmsprop. For triplet, contrastive, and Arcface loss, the distance to the nearest labeled example is used as the confidence score when selecting unlabeled examples for labeling. For softmax cross-entropy loss, the softmax probability score is used as the confidence score. Our proposed self-learning approach was run for 25 meta-iterations and results were averaged over 3 runs with a random selection of initially labeled examples.

4.1 Results

SVHN (Street View House Numbers) comprises 32x32 color images of house numbers. A single image can contain multiple digits, but only the digit in the center is considered for the label prediction. The proposed approaches are evaluated using 1000 labeled instances initially and use a selection percentage of 5% (i.e., in each meta-iteration of self-training, 5% of the remaining unlabeled examples are selected for labeling). Table 1 shows test accuracy for SVHN using all four losses, with random as well as pretrained weights, for the 1000-labeled, the self-learning, and the all-labeled setup.

The CIFAR-10 dataset comprises 32x32 RGB images of ten different object classes. The proposed semi-supervised approaches are evaluated using 4000 labeled instances initially, with a selection percentage of 5% for self-training. Table 2 shows accuracy on the standard test set for all losses using 4000-labeled, all-labeled and self-learning, for pretrained weights from Imagenet as well as random initial weights.

The Plant Village [9] dataset consists of plant leaves. It has 43,456 training and 10,849 test RGB images resized to 96x96 from the original format (256x256). It has 38 categories of species and diseases. A sample image for each class is shown in Figure 2. The proposed semi-supervised approaches are evaluated using 10 images per class as labeled instances initially, with a selection percentage of 2% in self-learning. Table 3 shows accuracy on test examples for all four losses using 380-labeled, all-labeled and self-learning, with random weight initialization and pretrained weights.

As we can see from the results for all three datasets, using pretrained weights generally results in substantial improvements over random initialisation. When comparing the four loss functions, cross-entropy emerges as the winner, with triplet loss often being second best. However, especially for small numbers of

Table 1: SVHN Test Accuracy %.

Pretrained	1000 Labels	Self-learning	73257 Labels
Cross-entropy loss			
No	75.81 ± 2.28	92.07 ± 0.35	95.72 ± 0.23
Yes	80.84 ± 0.74	**92.73 ± 0.52**	**96.10 ± 0.21**
Triplet loss [21]			
No	57.22 ± 1.81	64.69 ± 1.39	94.79 ± 0.06
Yes	**82.52 ± 2.14**	86.14 ± 1.11	95.12 ± 0.23
Contrastive loss [8]			
No	54.73 ± 0.57	62.80 ± 0.63	81.82 ± 2.29
Yes	79.46 ± 0.99	82.59 ± 0.31	93.41 ± 0.26
Arcface loss [6]			
No	68.33 ± 0.91	70.42 ± 1.59	93.74 ± 0.11
Yes	80.84 ± 0.21	82.01 ± 1.41	95.66 ± 0.31

Table 2: CIFAR10 Test Accuracy %.

Pretrained	4000 Labels	Self-learning	50000 Labels
Cross-entropy loss			
No	70.43 ± 1.43	79.15 ± 0.80	87.84 ± 0.39
Yes	**77.07 ± 0.91**	**83.33 ± 0.19**	**89.37 ± 0.49**
Triplet loss [21]			
No	68.35 ± 3.63	70.57 ± 1.17	86.54 ± 0.42
Yes	76.42 ± 2.19	78.36 ± 1.39	88.15 ± 0.36
Contrastive loss [8]			
No	34.90 ± 0.73	44.58 ± 1.67	71.16 ± 0.05
Yes	71.98 ± 0.95	76.58 ± 0.05	85.92 ± 0.32
Arcface loss [6]			
No	55.04 ± 1.36	69.54 ± 3.69	75.31 ± 0.24
Yes	74.76 ± 0.72	76.55± 1.80	87.76 ± 0.24

Fig. 2: Plant Village disease [9] dataset

Table 3: Plant village 96x96 Test Accuracy %.

Pretrained	380 Labels	Self-learning	43456 Labels
Cross-entropy loss			
No	45.78 ± 4.09	54.58 ± 2.65	98.24 ± 0.62
Yes	73.76 ± 1.70	**84.62 ± 1.2**	99.24 ± 0.08
Triplet loss [21]			
No	29.81 ± 2.59	33.16 ± 1.96	92.15 ± 1.63
Yes	**76.88 ± 0.36**	77.80 ± 1.15	99.02 ± 0.11
Contrastive loss [8]			
No	13.12 ± 1.56	16.35 ± 0.88	34.75 ± 3.20
Yes	30.22 ± 2.14	32.46 ± 2.65	45.66 ± 2.64
Arcface loss [6]			
No	54.85 ± 0.09	58.39 ± 3.61	98.11 ± 0.38
Yes	60.67 ± 0.13	71.80 ± 2.58	**99.32 ± 0.04**

labeled examples, triplet loss seems competitive with cross-entropy, outperforming it for two of the three datasets. This seems reasonable, as paying explicit attention to the similarities of particular instances may be more important when only a few labeled instances are available.

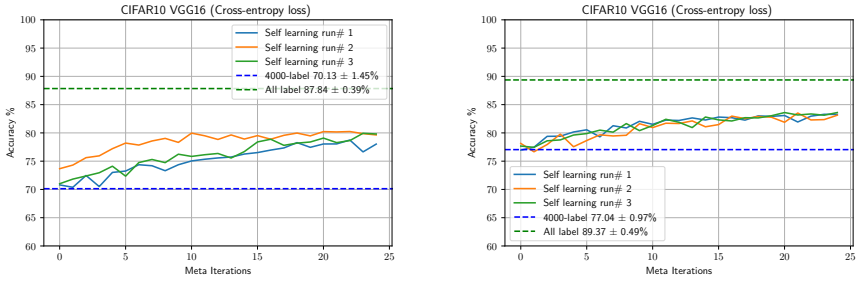
Comparing the three metric losses with each other, triplet loss generally outperforms the other two when using pretrained weights. On the other hand, when using random initial weights, none of the three losses seems to have a clear advantage over the others, except for the Plant dataset, where Arcface performs very well, even outperforming cross-entropy.

Figure 3 shows a comparison of self-learning using random weights and pretrained weights, across three different runs on CIFAR10, using softmax cross-entropy loss for 4000 initially labeled examples and 25 meta-iterations of self-learning. The accuracy curves show similar improvements for both scenarios, with the pretrained version starting from a higher initial accuracy level, and retaining this advantage over the 25 meta-iterations of self-learning.

In order to investigate the effect of self-learning on the embeddings, we visualize the embeddings obtained using all four loss functions. Figure 4 shows the output of TSNE [13] on embeddings of CIFAR10 test instances after training on 4000 labeled examples and after 25 meta-iterations of self-learning using all four losses. It is evident that self-learning improves class separation, with cross-entropy showing the most dramatic improvement, consistent with its high final accuracy.

5 Conclusions

In this paper, we have shown that transfer learning can be highly beneficial for semi-supervised image classification. In terms of loss functions, overall, cross-



(a) Self-learning with random weights (b) Self-learning with Imagenet weights

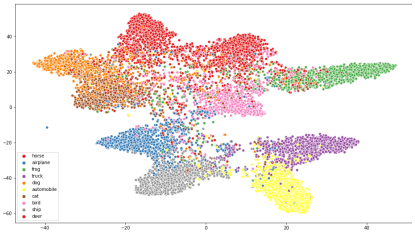
Fig. 3: CIFAR10 meta-iterations of self-learning using random and pretrained Imagenet weights.

entropy outperforms more specialised losses like triplet loss, contrastive loss, or Arcface loss. Still, for a small number of labels, triplet loss is very competitive.

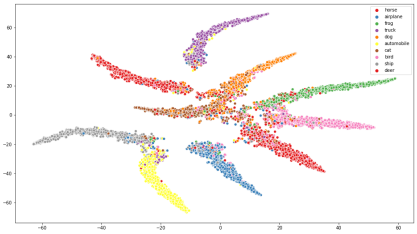
There are a number of directions for future work. Exploring combinations of well-performing loss functions, exploring alternatives to the label propagation scheme, and exploring connections to few-shot learning, are just a few obvious ones. Additionally, more lower-level engineering ideas, like mini-batch composition strategies as pointed out in [1], might help to further improve the performance of semi-supervised image classification.

References

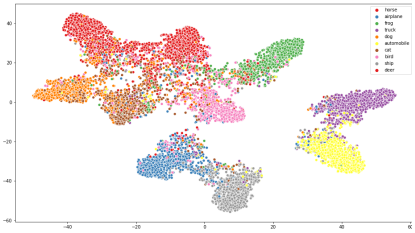
1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2020)
2. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "Siamese" time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence **07**(04), 669–688 (1993)
3. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning. The MIT Press (2006)
4. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 539–546. IEEE (2005)
5. Dai, Z., Yang, Z., Yang, F., Cohen, W.W., Salakhutdinov, R.R.: Good semi-supervised learning that requires a bad gan. In: Advances in Neural Information Processing Systems. pp. 6513–6523 (2017)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)



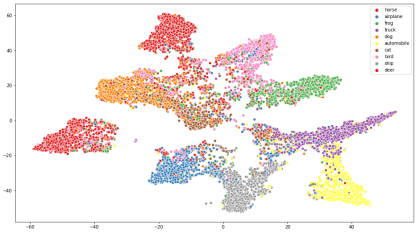
(a) Cross-entropy: initial training



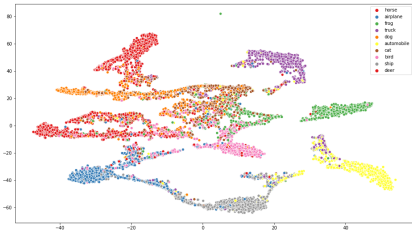
(b) Cross-entropy: after 25 meta-iterations



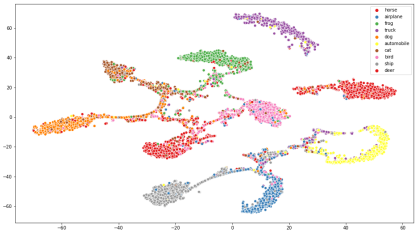
(c) Triplet loss: initial training



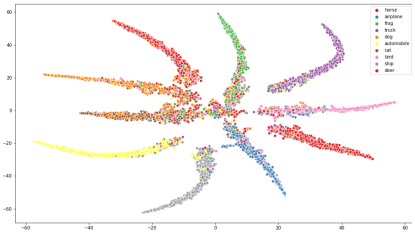
(d) Triplet loss: after 25 meta-iterations



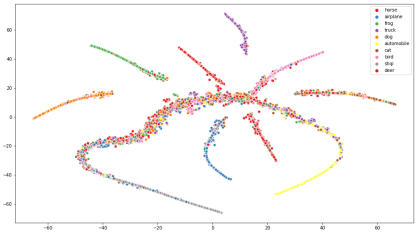
(e) Contrastive loss: initial training



(f) Contrastive loss: after 25 meta-iterations



(g) Arcface: initial training



(h) Arcface: after 25 meta-iterations

Fig. 4: TSNE Visualization of CIFAR10 embeddings for all losses after the first 4000 labeled examples and after 25-meta iterations of self-learning.

8. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 1735–1742. IEEE (2006)
9. Hughes, D., Salathé, M., et al.: An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060 (2015)
10. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML. vol. 3, p. 2 (2013)
11. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 212–220 (2017)
12. Maaløe, L., Sønderby, C.K., Sønderby, S.K., Winther, O.: Auxiliary deep generative models. In: Proceedings of the 33rd International Conference on Machine Learning-Volume 48. pp. 1445–1454 (2016)
13. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008)
14. Miyato, T., Maeda, S.i., Koyama, M., Ishii, S.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence **41**(8), 1979–1993 (2018)
15. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems. pp. 3546–3554 (2015)
16. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. In: 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1. vol. 1, pp. 29–36 (2005)
17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
18. Sahito, A., Frank, E., Pfahringer, B.: Semi-supervised learning using Siamese networks. In: Liu, J., Bailey, J. (eds.) AI 2019: Advances in Artificial Intelligence. pp. 586–597. Springer International Publishing, Cham (2019)
19. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Advances in Neural Information Processing Systems. pp. 1163–1171 (2016)
20. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems. pp. 2234–2242 (2016)
21. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 815–823 (2015)
22. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5265–5274 (2018)
23. Wei, X., Gong, B., Liu, Z., Lu, W., Wang, L.: Improving the improved training of wasserstein GANs: A consistency term and its dual effect. In: International Conference on Learning Representations (2018)
24. Weston, J., Ratle, F., Mobahi, H., Collobert, R.: Deep learning via semi-supervised embedding. In: Neural Networks: Tricks of the Trade, pp. 639–655. Springer (2012)