Introduction

Deep neural network-based image classifiers typically require a large amount of labelled data to perform well in classification tasks. A major problem is that acquiring labelled images in some domains is time-consuming and/or expensive. Such domains range from the medical field, where medical professionals may be needed for labelling, to bird spotting, where ornithologists are needed to ensure accurate data. Since collecting images is relatively easy, we often have an abundance of unlabeled images in comparison to labelled ones. These unlabeled images can be utilized to improve the performance of a deep network, which is useful when we only have a small amount of labelled data. Training a model with a mix of labelled and unlabeled data is referred to as semi-supervised learning (SSL). One approach in SSL is contrastive learning (CL), which learns to group similar data entries and distance them from the different ones [Hadsell et al., 2006]. Two other approaches are pseudo-labelling, which is when a model creates labelled data from unlabeled data by using a trained model, and consistency regularisation, which aims at making consistent predictions for the same, possibly augmented, input image [Sohn et al., 2020]. The two latter approaches is combined in *FixMatch* [Sohn et al., 2020], which is brought forward in this essay. Furthermore, I consider the papers of the SSL-inspired *DivideMix* [Li et al., 2020] and the CL-focused *SimCLR* [Chen et al., 2020].

Method

FixMatch [Sohn et al., 2020] combines consistency regularization and pseudo-labelling to train a model with a mix of labelled and unlabeled images. [Sohn et al., 2020] furthermore work with two sets of augmentations, weak and strong augmentations. The weak augmentations consist of flip-and-switch augmentations and for strong augmentations, they use augmentation methods proposed in previous work that changes the images to a larger extent.

The loss that is minimized in FixMatch, using standard SGD, consists of the sum of two terms, the supervised term l_s and the unsupervised term λl_u where λ is a hyper-parameter. l_s is calculated using the standard cross-entropy loss between the class label for data point x_i and the prediction made when x_i is weakly augmented. The unsupervised portion of the loss, l_u, is calculated as the cross-entropy between the label given to an unlabeled weakly augmented point u_i and the prediction of a strongly augmented version of the same point (u_i). Each unlabeled point is only included in the l_u term if the output value of the predicted label of the weakly augmented version of u_i is larger than a fixed threshold, τ. This prevents using unlabeled data that the model is uncertain about too early in the learning process.

DivideMix [Li et al., 2020] is a proposed model for training with noisy labelled data. The authors' motivation for studying the use of noisy labelled data is that it can be easier and cheaper to acquire compared to data sets with strictly correct labels. Their underlying idea of DivideMix is to identify the labels in the data that are likely to be noisy and use them as unlabeled data during training. To do this, they let the sample-wise loss in a network form a two-component Gaussian mixture model (GMM), where one GMM component is meant to capture the sample-wise loss distribution of clean labels and the other one to capture the loss

distribution of the noisy labels. At each epoch, a data point x_i is treated as either labelled or unlabeled data depending on how probable it is, according to the model from the previous epoch, to belong to the clean component of the GMM, thresholded by the hyper-parameter τ. Before any labelled/unlabeled splits, the network is trained for a few epochs ([Li et al., 2020] use 10 epochs on CIFAR-10) with all labels.

DivideMix uses two randomly initialized networks to split the data into labelled/unlabeled sets for each other at each epoch to combat confirmation bias. [Li et al., 2020] call this approach co-divide. During each epoch, they train on sampled mini-batches containing both labelled and unlabelled data points, one network at a time. Each data point in a mini-batch is augmented M number of times and the predictions are then averaged across the augmentations. A modified version of MixMatch [Berthelot et al., 2019] is used on the mini-batch containing labelled and unlabeled data. For the labelled data, [Li et al., 2020] use labels which are linear combinations of the true labels and the predicted labels, adjusted by the cleanliness of each label according to the other network. For the unlabeled set, they include and average the predictions made by the other network as well. The refined labels and averaged predictions are then sharpened before applying MixMatch and then updating the weights of the network in focus.

SimCLR [Chen et al., 2020] is a framework for CL, mapping images from input to the latent space, without the requirement of labelled data until it is fine-tuned afterwards. The model consists of a *base encoder* (e.g. ResNet-50) and a smaller neural network on top of that, referred to the *projection head*. For the projection head, [Chen et al., 2020] use a 2-layer MLP that maps to the latent space. This model learns to produce the same latent space for different augmentations of the same input data x_i. The augmentations consist of; cropping, colour distortions, and Gaussian blur applied in order with some randomness involved to get different augmentations.

The first step in SimCLR is to apply CL from the input space to the latent space using unlabeled images. At each update step, they randomly sample mini-batches of size N, augment each image with two different augmentations and get 2N augmented images to work with. The pair of augmented images, x_i' and x_i'', that originates from the same original image x_i becomes a positive pair and the augmentations of the other images in the mini-batch become a negative pair when coupled with either x_i' or x_i''. The contrastive loss function is calculated using all positive pairs and all negative pairs that are possible in the mini-batch. Both the base encoder and the projection head weights are updated but the projection head is discarded after training. The base encoder is then fine-tuned on labelled data and used as a classifier.

Comparison

The main difference between DivideMix and the other two approaches regarding the used data is that DivideMix is trained on data with noisy labels for all data points while the other use a mix of labelled and unlabeled data points. However, DivideMix uses the labels as a way to create unlabeled and labelled splits, and then learns in an SSL-fashion each epoch. This difference makes DivideMix the preferred model when we have noisy data and the other two when we have a mix of labelled and unlabeled data. Revisiting the problem of the costs

associated with gathering accurate data from the introduction, I can see how DivideMix, and other models learning with noisy labels, can be useful when amateurs have been used to label data that typically requires experts. If we for example only have unlabeled data, we can hire an amateur to label all the data somewhat accurately, cheaply, and use it with DivideMix since we expect some label noise, instead of only labelling a small portion that we can afford with a professional. This does not necessarily mean that this approach will lead to a DivideMix model that outperforms the other two, but it is something to consider in a project when hiring a professional isn't an option.

A common trick that the proposed models use is that they augment the images in some way during training, indicating that data augmentation is a powerful tool in semi-supervised learning with images. Their differences in the chosen augmentations also indicate that there isn't a one fits all augmentation that is universally used in all state-of-the-art ML research. [Chen et al., 2020] did a systematic study on pairs of transformations used (sequentially) in SimCLR which shows that they explored multiple approaches instead of deciding one immediately.

The robustness of the models is reflected by how much they tune the hyper-parameters between experiments. FixMatch demonstrates robustness in the experiments in [Sohn et al., 2020]. In FixMatch, they only tune the weight decay (out of the presented hyper-parameters) between CIFAR-10, CIFAR-100, SVHN and STL-10, but keep it constant between different ratios of unlabeled/labelled data for each experiment. DivideMix [Li et al., 2020] uses the same hyper-parameters between CIFAR-10 and CIFAR-100 apart from a loss weight hyper-parameter, which also varies depending on the asymmetry of the label noise. This may cause problems in real-world applications where the label noise asymmetry is unknown, thus requiring more tuning for better performance. [Chen et al., 2020] use a default setting for the hyper-parameters used in SimCLR but tune the hyperparameters on the additional transfer learning tasks.

A difference between SimCLR and the other two is that in order to use SimCLR, one has to train it in two steps, first on unlabeled data, and then fine-tune it with labelled data. The other two learn in an unsupervised fashion and supervised at the same time. The benefit of doing both simultaneously is that it becomes easier to baby-sit the error rate learning curve from start to finish. With SimCLR, one is required to fine-tune a copy of the network between intervals in order to analyze the accuracy of the model at each update step.

The differences in data sets used, unlabeled/labelled splits, and performance metric used makes it difficult to strictly compare them in terms of performance. However, FixMatch shows impressive results on data with very few labels per class across multiple data sets. FixMatch furthermore slightly outperforms SimCLR on ImageNet with 10% of the available labels with ResNet-50. SimCLR achieves a top-5 accuracy of 87.8% and FixMatch achieves 89.13% on the same metric. While FixMatch performed better in this specific comparison, DivideMix showed that the performance can be further improved to 92.6% by increasing the hidden layer widths.

References

[Berthelot et al., 2019] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059.

[Chen et al., 2020] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

[Hadsell et al., 2006] Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

[Li et al., 2020] Li, J., Socher, R., and Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

[Sohn et al., 2020] Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*.